

# QFlow: Um Sistema com Garantia de Isolamento e Oferta de Qualidade de Serviço para Redes Virtualizadas\*

Diogo Menezes Ferrazani Mattos e Otto Carlos Muniz Bandeira Duarte

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

**Resumo.** A técnica de virtualização de redes é essencial para prover o compartilhamento dos recursos físicos e permitir a implantação de inovações no núcleo das redes. Contudo, garantir isolamento entre redes virtuais e assegurar a oferta de Qualidade de Serviço (QoS) ainda são desafios tecnológicos. Este artigo propõe o sistema QFlow para isolar o uso de recursos e prover QoS em uma arquitetura híbrida de redes virtuais que combina uma ferramenta de virtualização, como o Xen, para implementar o plano de controle, e a interface de programação de aplicação (API) OpenFlow, para o controle do mecanismo de encaminhamento de pacotes. Os dois principais elementos do QFlow são o controlador de uso de recursos e a aplicação de mapeamento de parâmetros de QoS. O sistema QFlow permite oferecer diferenciação de serviços inter-redes virtuais distintas e intrarrede virtual, garantindo recursos a diferentes fluxos de uma rede virtual. Um protótipo foi desenvolvido e a avaliação do sistema proposto mostra que o controlador de recursos é capaz de atender aos requisitos de qualidade de serviço das redes virtuais e, também, é capaz de melhor redistribuir os recursos ociosos da rede quando comparado a outras técnicas.

**Abstract.** The network virtualization is an essential technique to provide physical resources sharing and to allow implementation of innovations in the core of networks. Nevertheless, isolation between virtual networks and Quality of Service (QoS) provisioning are still technological challenges. In this paper, we propose the QFlow system to isolate resource usage and to provide QoS guarantees in a hybrid virtual network architecture, which combines a virtualization tool, such as Xen, to implement the control plane, and the Application Programming Interface (API) OpenFlow to control the packet forwarding engine. QFlow main elements are the resource usage controller and QoS parameters mapping application. The QFlow system offers inter-network service differentiation and intra-network QoS provisioning, guarantying resources to different flows of a virtual network. We developed a prototype and the obtained results show that the resource controller is able to meet the requirements of quality of service of the virtual networks, and performs also better than other techniques to redistribute idle network resources.

## 1. Introdução

A virtualização de redes é uma técnica promissora para o desenvolvimento de propostas de novas arquiteturas para Internet [Mattos et al. 2011b, Feamster et al. 2007, Moreira et al. 2009]. A virtualização desacopla a função executada por um sistema de sua realização física e, no contexto de virtualização de redes, conduz à possibilidade de

---

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

inovar no núcleo da rede, pois a rede física passa a ser compartilhada por múltiplas fatias chamadas de redes virtuais, as quais possuem pilhas de protocolos e arcabouços de controle próprios. As redes virtuais devem ser isoladas e possuem requisitos de qualidade de serviço (QoS) diferenciados [Keller and Rexford 2010, McKeown et al. 2008, Kim et al. 2010].

A virtualização de redes pode ser alcançada através de ferramentas de virtualização de computadores, como é o caso do Xen [Egi et al. 2008], em que as máquinas virtuais se comportam como roteadores; ou através da programação de elementos de rede, como é feito no OpenFlow [McKeown et al. 2008], em que o controle da rede, executado por um nó independente e centralizado, interage com os elementos encaminhadores através de uma interface de programação de aplicação (API). Contudo, essas ferramentas apresentam desafios para a provisão de qualidade de serviço e para o isolamento do uso de recursos compartilhados.

Este artigo apresenta o QFlow, um sistema de controle de qualidade de serviço e de isolamento de recursos, como CPU, memória e banda, em redes virtuais. O sistema QFlow baseia-se na arquitetura híbrida de virtualização de redes usada no sistema XenFlow [Mattos et al. 2011b], que combina a plataforma de virtualização de computadores Xen com a interfaces de programação de aplicação OpenFlow. As principais contribuições do sistema QFlow são: i) a garantia de isolamento e a oferta de qualidade de serviço através de um eficiente controle dos recursos disponíveis; ii) um mecanismo de mapeamento dos parâmetros de Qualidade de Serviço, definidos para cada rede virtual, em recursos do plano de encaminhamento de dados; e iii) um algoritmo de redistribuição de recursos ociosos na rede física entre as redes virtuais, de forma que os recursos sejam redistribuídos proporcionalmente à necessidade e à prioridade de cada rede virtual.

As principais propostas para prover QoS são baseadas no provisionamento exagerado de recursos da infraestrutura física da rede e na configuração manual de parâmetros de QoS nos dispositivos físicos [McIlory and Sventek 2006, Kim et al. 2010]. O sistema proposto, diferentemente, monitora o uso de recursos por cada rede virtual e, com base nesses dados, calcula a redistribuição dos recursos ociosos na infraestrutura física. A ideia chave do QFlow é um controlador de recursos que define valores mínimo e máximo de cada fila e, assim, oferece QoS inter-redes, e também QoS intrarrede. Um protótipo do sistema QFlow foi desenvolvido e avaliado. Os resultados mostram que o algoritmo de redistribuição de recursos reduz a ociosidade da rede, quando comparado a outras técnicas, e atribui a banda reservada para cada rede virtual proporcionalmente ao valor contratado por cada rede. O sistema permite, então, o atendimento dos acordos de nível de serviço das redes virtuais, assim como otimiza o uso de recursos da rede física.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta o modelo de virtualização híbrido de redes. O sistema proposto é detalhado na Seção 4. A avaliação do sistema e os seus resultados são apresentados na Seção 5. A Seção 6 conclui o artigo.

## 2. Trabalhos Relacionados

A tecnologia de virtualização permite a abordagem pluralista de diversas redes virtuais executando sobre um mesmo substrato físico. Há propostas de definições de redes por *software* para atender características “sob medida” (*software defined networks*). No entanto, o isolamento e a provisão de QoS em redes virtualizadas ainda são desafios para as principais plataformas de virtualização [Fernandes and Duarte 2011].

Houidi *et al.* propõem um sistema adaptativo para o provimento de redes virtu-

ais [Houidi et al. 2010]. A ideia central é prover recursos sob demanda a redes virtuais, após da verificação da degradação do serviço prestado por cada rede virtual ou após a falha de algum recurso. Para tanto, o sistema utiliza um mecanismo multiagentes distribuído na infraestrutura física para negociar a requisição, realizar a adequação dos recursos e sincronizar os nós fornecedores e as redes virtuais. Outra proposta, o OMNI (*OpenFlow Management Infrastructure*) [Mattos et al. 2011a], também se baseia em um sistema multiagentes para prover qualidade de serviço a redes OpenFlow [McKeown et al. 2008]. Entretanto, as ações tomadas pelos agentes OMNI restringem-se à migração de fluxos em uma rede OpenFlow. Kim *et al.*, propõem um sistema de mapeamento de parâmetros de QoS em recursos disponíveis em comutadores OpenFlow, tais como filas e limitadores de taxa [Kim et al. 2010]. O objetivo desse sistema é prover QoS em cenários em que a infraestrutura física é compartilhada por redes virtuais com diferentes demandas. Contudo, o controle dos parâmetros de QoS e o mapeamento são centralizados no nó controlador da rede OpenFlow.

Uma outra forma de prover QoS a redes virtuais é apresentada por McIlroy e Sventek [McIlroy and Sventek 2006]. A proposta consiste em uma nova arquitetura de roteador, na qual o roteador é composto por diversas máquinas virtuais, chamadas *Routelets*. Cada *Routelet* é isolado dos demais e os seus recursos são garantidos e limitados. Nessa arquitetura, os fluxos sensíveis a QoS são roteados pelos *Routelets*, cuja prioridade de acesso a recursos do substrato define a qualidade de serviço de cada fluxo que encaminha. Contudo, o encaminhamento de pacotes nessa proposta é realizado pelas máquinas virtuais, o que limita o desempenho de encaminhamento dos *Routelets*.

Fernandes e Duarte [Fernandes and Duarte 2010, Fernandes and Duarte 2011] propõem também mecanismos de controle e gerenciamento de redes virtuais. Os mecanismos propostos realizam o controle local, ou seja, restrito ao nó provedor do ambiente de virtualização Xen [Egi et al. 2008]. As propostas baseiam-se no isolamento dos recursos usados por cada ambiente virtual no domínio de controle Xen e na diferenciação dos serviços providos por cada rede virtual. A proposta se restringe à plataforma de virtualização Xen e depende da marcação de pacotes.

Wang *et al.* apresentam o OpenFlow como uma ferramenta de gerência de redes [Wang et al. 2011]. Nessa proposta, o OpenFlow é usado como uma plataforma de balanceamento de cargas, baseado em comutadores de baixo custo, que multiplexam requisições entre diferentes réplicas de um servidor. A solução proposta usa pesos para fragmentar o espaço de endereçamento IP dos clientes entre os servidores. Assim, de acordo com o IP do cliente, é possível identificar qual réplica deve atendê-lo. A proposta, no entanto, não garante a reserva de recursos, nem a qualidade de serviço dos fluxos.

Hao *et al.*, por sua vez, apresentam a infraestrutura VICTOR (*Virtually Clustered Open Router*) [Hao et al. 2009] que se baseia em criar um agrupamento de *datacenters*, através de uma infraestrutura de rede virtualizada. A ideia central dessa proposta é usar o OpenFlow como a infraestrutura básica da rede de *datacenters*, de modo a permitir que ao migrar uma máquina virtual de uma localização física para outra, seja possível reconfigurar os caminhos na rede. Essa proposta otimiza o uso da rede realizando migrações de servidores, porém não garante a qualidade de serviço dos fluxos e, também, não isola o uso de recursos decorrente dos diferentes servidores virtualizados.

Uma proposta de sistema híbrido de virtualização de redes, com foco no roteamento, é o RouteFlow [Nascimento et al. 2011]. O principal objetivo do RouteFlow é permitir que protocolos legados de roteamento exerçam o controle centralizado de uma infraestrutura de rede OpenFlow. O RouteFlow baseia-se em replicar a topologia física

de uma rede OpenFlow em um ambiente virtual. Nesse ambiente virtual composto por máquinas virtuais, cada uma representa um nó da rede física OpenFlow e executa protocolos de roteamento. O ambiente virtual se comunica com a rede OpenFlow através do controlador da rede. O controlador da rede, por sua, é um nó centralizado responsável por programar os planos de dados de todos os comutadores OpenFlow. No entanto, essa proposta não isola os recursos, nem define parâmetros de QoS para cada rede.

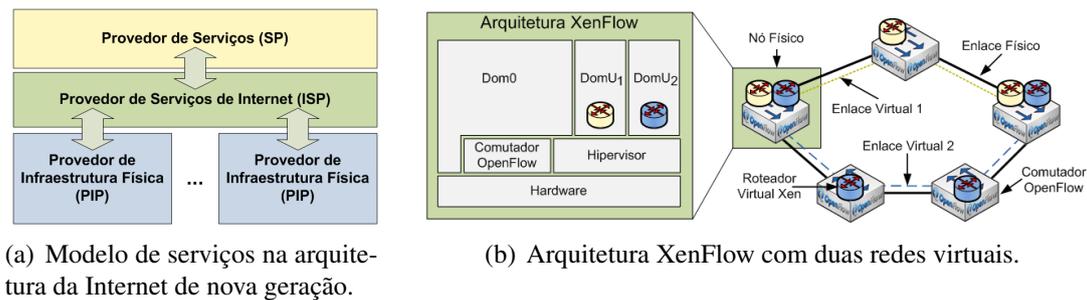
O XenFlow [Mattos et al. 2011b] é outro sistema híbrido de virtualização de redes que combina as ferramentas de virtualização de computadores Xen e a de programação de redes OpenFlow. O objetivo do XenFlow é a construção de um sistema em que os protocolos de roteamento são executados em máquinas virtuais Xen e o encaminhamento dos pacotes é realizado por um comutador OpenFlow. Uma das principais vantagens do XenFlow é a provisão da funcionalidade de migração ao vivo. No entanto, o sistema XenFlow não isola as redes virtuais no plano de encaminhamento dos pacotes, assim como não provê primitivas de qualidade de serviço para as redes virtuais.

As principais inovações do QFlow em relação a essas propostas são o isolamento de recursos de cada rede virtual no plano de encaminhamento, a introdução de um mecanismo de controle de QoS e um algoritmo de redistribuição dos recursos excedentes de acordo com a prioridade e com o uso de cada rede virtual.

O sistema QFlow proposto neste artigo se baseia no modelo híbrido do XenFlow de virtualização de redes, combinando as ferramentas de virtualização de computadores Xen e a de programação de redes OpenFlow. No entanto, ao contrário do XenFlow, o QFlow garante o isolamento dos recursos compartilhados pelo roteadores virtuais no domínio privilegiado de controle do Xen, tornando-o robusto a ataque de negação de serviço, e também oferece garantias de qualidade de serviço. O sistema QFlow realiza o compartilhamento eficiente dos recursos ociosos e suporta o atendimento a acordos de nível de serviço (SLAs – *Service Level Agreements*), como por exemplo uma garantia mínima de banda destinada a um roteador virtual.

### 3. O Modelo Híbrido de Virtualização de Redes

No modelo atual da Internet, os provedores de serviços (*Service Provider - SP*) fornecem serviços através da Internet e os provedores de serviço de Internet (*Internet Service Provider - ISP*) fornecem conectividade à Internet aos seus clientes. Contudo, projeções para os novos modelos de negócio da Internet apontam para necessidade de desacoplar as funções de fornecer e manter os equipamentos físicos de rede das funções de prover conectividade fim-a-fim e serviços na Internet, criando a necessidade de um novo ator, o provedor de infraestrutura física (*Physical Infrastructure Provider - PIP*) [Carapinha and Jiménez 2009]. A principal proposta para realizar o desacoplamento de funções é a virtualização de redes, pois permite que um ISP execute múltiplos serviços fim-a-fim sobre equipamentos de rede de diferentes provedores de infraestrutura [Feamster et al. 2007], como mostrado na Figura 1(a). Nesse novo cenário, os acordos de nível de serviço (*Service Level Agreements - SLAs*) estabelecidos entre os ISPs e seus clientes devem ser mapeados na rede física. Assim, para garantir uma determinada qualidade de serviço a uma rede virtual, é necessário obter garantias tanto do provedor de rede virtual (ISP) quanto do provedor de infraestrutura (PIP), como também é necessário um mapeamento entre os parâmetros estabelecidos pelo ISP e os recursos fornecidos pelo PIP. O sistema QFlow proposto permite a garantia e o gerenciamento de QoS através do plano de encaminhamento de dados e também realiza o mapeamento dos parâmetros de QoS do plano de controle da rede virtual para o plano de dados. Portanto, o sistema QFlow oferece garantias de qualidade de serviço atendendo a esse novo modelo de negócios.



**Figura 1. Modelo de virtualização de redes com XenFlow.**

O modelo de virtualização aplicado nesse artigo é o modelo híbrido XenFlow [Mattos et al. 2011b]. O modelo XenFlow de virtualização de redes permite que um provedor de infraestrutura forneça roteadores virtuais aos provedores de serviço de Internet e esses, por sua vez, usem os roteadores virtuais para prover conectividade fim-a-fim a seus clientes. Nesse contexto, um roteador virtual é um ambiente virtualizado com interfaces semelhantes às de uma máquina real e executa sobre o substrato físico compartilhado com outros roteadores virtuais, como mostra a Figura 1(b). O modelo de virtualização XenFlow é dividido em dois componentes principais: o plano de controle Xen e o plano de dados OpenFlow. O plano de controle Xen é o ambiente virtualizado ao qual o ISP contratante tem acesso e executa os seus protocolos de roteamento. Já o plano de dados OpenFlow é a estrutura de encaminhamento compartilhada por todos os roteadores virtuais hospedados sobre um mesmo substrato físico. A interação entre o plano de controle e o plano de dados restringe-se à informação das rotas calculadas no plano de controle e repassadas ao plano de dados. Com isso, as políticas de QoS definidas no plano de controle não são refletidas no real encaminhamento dos dados no plano de dados.

Este artigo propõe o sistema QFlow que controla o uso dos recursos do plano de dados do modelo de virtualização XenFlow. Assim, o sistema proposto garante o completo isolamento das redes virtuais e, ao mesmo tempo, possui o total controle dos recursos físicos garantindo oferta de qualidade de serviço por rede virtualizada. Os recursos controlados são o processamento, a memória e a banda passante de cada roteador virtual. O controle de processamento é feito diretamente pelo escalonador do Xen, limitando-se o quanto da CPU é atribuído para cada roteador virtual. O controle de memória é realizado limitando-se o número de rotas exportadas para o plano de dados. O controle de banda é realizado por dois algoritmos que garantem uma taxa mínima para oferta de qualidade de serviço e uma taxa máxima para otimizar o uso de recursos de rede, redistribuindo a capacidade ociosa dos enlaces entre os roteadores virtuais. O controle das taxas mínima e máxima de cada rede virtual é realizado através da instanciação de filas associadas às interfaces físicas e do mapeamento das interfaces de cada roteador virtual nas filas de cada interface física.

#### 4. O Sistema Proposto

O QFlow é um sistema que provê isolamento e qualidade de serviço em redes virtualizadas. O QFlow garante isolamento das redes virtuais ao oferecer e garantir recursos mínimos para cada uma das redes virtuais. Assim, uma determinada rede virtual não pode consumir os recursos de outra rede virtual evitando ataques de negação de serviço (*Denial of Service* – DoS). O sistema QFlow oferece também qualidade de serviço através do mapeamento dos parâmetros de acordos de nível de serviço (*Service Level Agreements* – SLAs) do plano de controle para os parâmetros do plano de dados das redes virtuais,



adotada no QFlow é uso de filas de encaminhamento associadas aos roteadores virtuais com políticas de controle do tipo HTB (*Hierarchical Token Bucket*).

A arquitetura do sistema QFlow, mostrada na Figura 2, é um computador pessoal executando a plataforma de virtualização Xen e a matriz de comutação de pacotes baseada na interface de programação de aplicação (*Application Programming Interface – API*) OpenFlow. As máquinas virtuais Xen executam protocolos de roteamento e comportam-se como roteadores virtuais. Neste trabalho, a plataforma de roteamento extensível XORP (*eXtensible Open Router Platform*) [Handley et al. 2003] permite que o roteador virtual execute os principais protocolos de roteamento e, ainda, permite a inclusão de novos protocolos. O roteador virtual executa o módulo *QFlow Client* que verifica as atualizações na sua tabela de rotas e na sua tabela ARP (*Address Resolution Protocol*). A tabela de rotas é a estrutura de dados que armazena as informações das rotas, calculadas pelos protocolos de roteamento executando no roteador virtual, e a tabela ARP é aquela que armazena o mapeamento dos IPs em endereços MAC conhecidos pelo roteador virtual. As informações coletadas na máquina virtual são, então, encaminhadas para o módulo *QFlow Server* que executa no Domínio 0. O módulo *QFlow Server* gera resumos das informações de todos os roteadores virtuais e os repassa para a aplicação *QFlow App*. A aplicação *QFlow App* executa sobre o controlador NOX do OpenFlow, instanciado, também, no Domínio 0 de cada nó QFlow.

A principal inovação do QFlow em relação às outras propostas é a introdução do módulo de controle de uso de recursos por cada rede virtual e do módulo de controle de filas. O plano de dados do roteador QFlow é um comutador por *software* Open vSwitch [Pfaff et al. 2009] com suporte ao controle pela API OpenFlow. Vale ressaltar que no QFlow todo processo de comunicação entre módulos é criptografado e autenticado usando o esquema de distribuição de chaves públicas (*Public Key Infrastructure – PKI*) e o padrão SSL 3.0 (*Secure Socket Layer*).

#### **4.1. O Controlador de Uso de Recursos**

Os três recursos controlados pelo QFlow são o processamento, a memória e a banda passante por interface de rede usados por cada roteador virtual. A seguir é explicado como o sistema controla cada um dos recursos citados.

##### **O Controle de Processamento**

O controle do processamento de cada roteador virtual é executado de forma estática. Para tanto, o QFlow usa os parâmetros do escalonador do Xen para definir a prioridade de execução de cada roteador virtual em relação aos demais e o quanto de tempo contínuo de processamento cada roteador pode receber. O uso dos mecanismos de escalonamento nativos do Xen é suficiente para garantir o isolamento na carga de processamento de cada roteador virtual, como mostrado em [Fernandes et al. 2010]. Contudo, o encaminhamento de pacotes pelo Domínio 0 pode acarretar uma sobrecarga de processamento no Domínio 0 que não é controlada pelos parâmetros do escalonador do Xen [Fernandes and Duarte 2011]. Nesse sentido, a limitação da carga de processamento por roteador virtual no Domínio 0 é realizada através da restrição da capacidade de encaminhamento de pacotes, através do controle de banda associado aos roteadores virtuais.

##### **O Controle de Memória**

O sistema QFlow apresenta duas áreas de memória distintas. A primeira é dedicada ao roteador virtual e suas aplicações, representada pela área de memória da máquina virtual Xen. A segunda é uma área de memória compartilhada no Domínio 0. O isolamento na área de memória dedicada é realizado pelos próprios mecanismos do Xen.

Entretanto, no Domínio 0 há a necessidade de outros mecanismos para limitar o uso de memória de cada roteador virtual.

No Domínio 0, um roteador virtual consome memória para armazenar uma cópia de sua tabela de rotas e de sua tabela ARP, além de consumir memória para armazenar os seus fluxos instanciados na tabela de fluxos do OpenFlow. Assim, o controle de memória para cada roteador é realizado limitando-se o número máximo de entradas que um roteador virtual pode ter na sua tabela de rotas exportada para o Domínio 0, assim como limita-se também o número máximo de fluxos associados a um único roteador virtual. Nesses casos, para evitar perda de pacotes por causa da falta de rotas ou de fluxos no Domínio 0, são adicionadas rotas e fluxos padrões que forçam todo novo fluxo a ser encaminhado pelo roteador virtual ao invés de ser encaminhado pelo plano de dados.

### O Controle de Banda

O controle da banda de cada roteador virtual é realizado através da associação das máquinas virtuais a filas na matriz de encaminhamento OpenFlow. O OpenFlow permite a criação de filas isoladas, em cada interface de rede. Cada fila tem dois limites definidos: o mínimo reservado e o máximo permitido. O mínimo reservado é o limite mínimo garantido para aquela fila no compartilhamento do enlace. O limite máximo é o valor máximo que a taxa de transmissão daquela fila pode assumir.

O mecanismo de mapeamento dos fluxos de cada roteador virtual para a fila correspondente é realizado pela aplicação *QFlow App*, que mapeia as rotas em fluxos e, também, ao definir a interface de saída para o fluxo, associa o fluxo à fila dedicada ao roteador virtual pelo qual o fluxo foi encaminhado. Contudo, o controle de filas exercido pelo OpenFlow é estático e não considera a prioridade nas filas para redistribuir a capacidade excedente do enlace. Dessa forma, uma das contribuições deste artigo é um controlador de filas que executa a redistribuição da capacidade ociosa excedente do enlace de acordo com a prioridade de cada fila e inversamente proporcional a quanto a fila está usando além da capacidade contratada.

### O Controlador de Filas

O controlador de filas é responsável pela redistribuição da capacidade ociosa do enlace entre as filas instanciadas, a qual é feita em dois níveis. O primeiro caso é quando a capacidade total do enlace é maior do que o somatório de todos os limites mínimos das filas associadas àquele enlace. Nesse caso, o controlador age sobre o limite mínimo das filas. O segundo caso é quando cada fila possui a sua capacidade mínima definida, porém o uso total do enlace é menor do que o somatório dos limites mínimos reservados para as filas. Nesse caso, o controlador age sobre o limite máximo de cada fila. Em ambos os casos, a redistribuição considera a prioridade de cada rede e, no QFlow, a prioridade de uma rede é o valor relativo da banda contratada pela dada rede dividido pelo somatório dos valores de banda contratados por todas as redes que compartilham o enlace. O controlador mede o uso de cada fila regularmente a cada  $t_{med}$  segundos, armazena os últimos  $k_{med}$  valores de uso para cada fila e, sobre esses dados, calcula a redistribuição da capacidade ociosa da fila.

#### O Controle do Limite Mínimo Garantido

O controle do limite mínimo redistribui os recursos não atribuído a filas segundo dois critérios:

- o roteador virtual que contrata uma maior capacidade de encaminhamento tem **maior peso** na redistribuição de recursos ociosos;
- o roteador virtual que está usando relativamente mais recursos do que contratou

tem **menor peso** na redistribuição de recursos ociosos.

A redistribuição dos recursos ociosos entre as filas é proporcional aos pesos calculados para cada fila, como é mostrado no Algoritmo 1. A entrada do algoritmo de controle é dada pelas seguintes variáveis:

- *Porta*: representa o conjunto de filas associado a uma dada interface de rede;
- *LimMin*: vetor que representa os valores mínimos de transmissão garantidos no OpenFlow para cada uma das filas;
- *Carga*: vetor que representa os valores médios de carga para cada fila da interface controlada no momento em que o algoritmo está sendo aplicado.

A ideia central do Algoritmo 1 é calcular um valor relativo  $\alpha$  que representa o quão longe o valor da carga atual da fila está do valor mínimo contratado e garantido por SLA. Se  $\alpha$  é positivo, a carga atual da fila é menor do que o mínimo contratado. Caso seja negativo, a carga atual é maior que o mínimo contratado. Sendo assim,  $\alpha$  mede um percentual de recursos usados por cada fila além do mínimo contratado. O valor de  $\alpha$ , então, assume tanto valores positivos quanto negativos. O valor de  $\alpha$  é então mapeado para o intervalo de  $[0, 1]$  e passa a ser denominado  $\gamma$ . Os valores negativos de  $\alpha$  são mapeados para o intervalo  $[0, 0,5)$ , os positivos, no intervalo  $(0,5, 1]$  e  $\alpha = 0$  é mapeado em 0,5. Com posse dos valores de  $\gamma$  para todas as filas associadas a uma dada interface, os pesos de redistribuição são calculados através da normalização dos valores  $\gamma$ , de forma que a soma de todos os pesos das filas de uma interface seja sempre igual a 1.

---

**Algoritmo 1:** Cálculo dos pesos de redistribuição de recursos nas filas.

---

**Entrada:**  $Porta, LimMin[], Carga[]$   
**Saída:**  $Pesos[]$

- 1  $Zerar(\alpha[]); Zerar(\gamma[]);$
- 2 **para**  $fila \in Porta$  **faça**
- 3      $\alpha[fila] = \frac{LimMin[fila] - Carga[fila]}{\sum_{i \in Porta} LimMin[i]}$
- 4 **fim**
- 5 **para**  $fila \in \alpha$  **faça**
- 6     **se**  $(\alpha[fila] > 0)$  **então**  $\gamma[fila] = 0,5 \cdot \frac{\alpha[fila]}{Max(\alpha)} + 0,5$
- 7     **se**  $(\alpha[fila] = 0)$  **então**  $\gamma[fila] = 0,5$
- 8     **se**  $(\alpha[fila] < 0)$  **então**  $\gamma[fila] = 0,5 \cdot \frac{Min(\alpha - \alpha[fila])}{Min(\alpha)}$
- 9 **fim**
- 10  $total_{\gamma} = \sum_{i \in \gamma} i$
- 11 **para**  $fila \in \gamma$  **faça**
- 12      $Pesos[fila] = \frac{\gamma[fila]}{total_{\gamma}}$
- 13 **fim**

---

### *O Controle do Limite Máximo Permitido*

O controle do limite máximo permitido para cada rede virtual tem como objetivo otimizar o uso do enlace de acordo com a prioridade de cada rede virtual. O cálculo do valor máximo permitido é realizado somando-se a carga média de todas as filas em um enlace e, após, subtraindo-se esse valor da capacidade do enlace. O valor resultante é a capacidade que deve ser redistribuída. A redistribuição dada por

$$max_{fila} = min_{fila} + Capacidade\ Ociosa * \frac{min_{fila}}{\sum_{i \in Porta} min_i} \quad (1)$$

garante que os recursos não utilizados por uma fila serão atribuídos às demais filas proporcionalmente ao mínimo contratado por cada uma.

### A Provisão de QoS

A provisão da qualidade de serviço no sistema QFlow é realizada pelo controle de banda. A banda não utilizada do enlace é redistribuída de acordo com a prioridade de cada rede virtual. Além disso, dependendo de como é feito o mapeamento entre redes virtuais e filas, é possível definir-se duas modalidades diferentes para aplicar a QoS em redes virtuais:

- **Qualidade de Serviço Inter-rede** é quando associa-se cada rede virtual a uma única fila. Nesse caso, os parâmetros de QoS são apenas definidos para a rede virtual, permitindo diferenciar os serviços de uma rede virtual em relação aos serviços de outra;
- **Qualidade de Serviço Intranrede** é quando associa-se um conjunto de filas a um único roteador virtual. Cada fila tem os seus próprios parâmetros de QoS e, dependendo da configuração de cada roteador virtual, um determinado grupo de fluxos é mapeado em cada uma das filas desse roteador virtual. Esse esquema de mapeamento de conjuntos de fluxos em filas permite que seja reservada uma determinada banda para fluxos mais sensíveis dentro de uma mesma rede virtual. O caso da QoS intranrede é uma generalização do QoS interrede para um conjunto de fluxos e, portanto, não é excludente com o primeiro caso.

O mapeamento de fluxos em filas é realizado pela aplicação *QFlow App*, que executa sobre o NOX. No entanto, a definição de qual fila é dedicada a cada roteador virtual e, no caso intranrede, quais características dos fluxos são consideradas para mapear cada fluxo em uma fila são configuradas no módulo *QFlow Server* e comunicadas para a *QFlow App*.

## 5. Resultados

O protótipo desenvolvido se serve do Xen 4.0 para prover os planos de controle e do Open vSwitch 1.2.2 para prover a função de encaminhamento no plano de dados do sistema. O Open vSwitch [Pfaff et al. 2009] é configurado para ser usado pelo controlador NOX [Gude et al. 2008] do OpenFlow. A aplicação que realiza a separação de planos e o direcionamento dos pacotes para as filas adequadas no plano de encaminhamento de dados foi escrita em Python e executa sobre o NOX. O controlador de qualidade de serviço, o qual realiza os ajustes dos valores máximo e mínimo de cada fila, foi escrito em Python e interage diretamente com as interfaces de configuração do Open vSwitch. As ferramentas Iperf<sup>1</sup> e Tcpcdump<sup>2</sup> foram usadas para realizar as medidas de desempenho do sistema.

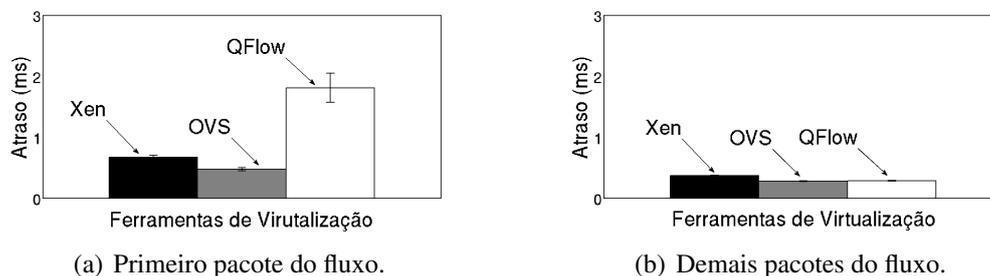
Quatro computadores pessoais compõem o cenário dos experimentos: um executa o protótipo QFlow ligando-se a outros três. Esse computador com QFlow tem a função de receber/encaminhar pacotes para outros dois computadores que geram ou recebem tráfego. O quarto computador controla o ambiente de testes. Todos os computadores possuem processadores Intel Core 2 Quad 2.4 GHz e 4 GB de memória RAM. Cada computador possui, no mínimo, 3 interfaces de rede sendo que todas são configuradas para

<sup>1</sup><http://iperf.sourceforge.com>.

<sup>2</sup><http://www.tcpdump.org>.

funcionarem a 100 Mb/s, uma vez que havia também interfaces de 1 Gb/s. O computador que desempenha a função de encaminhador e executa o sistema QFlow hospeda três máquinas virtuais que realizam a função de roteador. Cada máquina virtual é configurada com uma CPU virtual, 128 MB de memória RAM e executa o Debian Linux 2.6-32-5. As máquinas virtuais executam os protocolos de roteamento através da plataforma XORP [Handley et al. 2003], contudo, durante os testes, foram configuradas rotas estáticas. Os resultados apresentados nessa seção são médias de 10 rodadas de cada experimento, com intervalo de confiança de 95%.

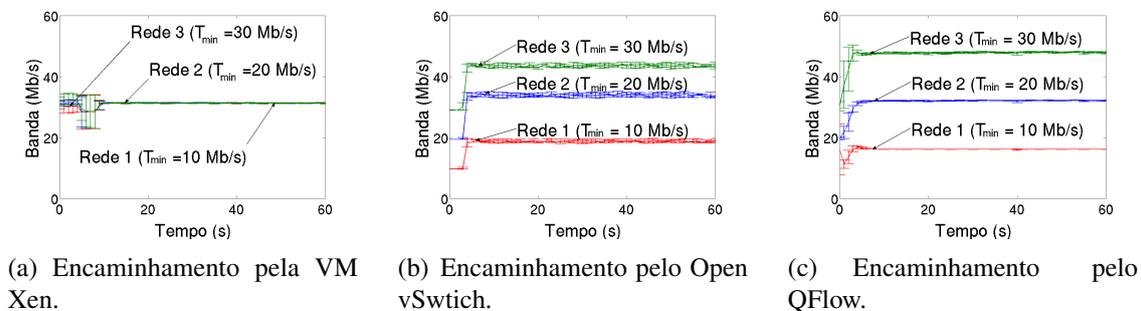
O primeiro experimento tem como objetivo avaliar o atraso introduzido pelo sistema QFlow no encaminhamento dos pacotes. A Figura 3 compara o atraso introduzido pelo roteamento QFlow com o atraso do Open vSwitch atuando como comutador, referenciado como OVS, e com o atraso quando a função de roteamento é executada na máquina virtual Xen, referenciado como Xen. Neste experimento, o Open vSwitch foi configurado para somente comutar os pacotes entre as interfaces do nó encaminhador, agindo semelhante a uma *bridge*; o roteamento pelo Xen foi configurado para os pacotes serem roteados pela máquina virtual e, por questão de justiça, o encaminhamento dos pacotes da interface de rede física para a interface de rede da máquina virtual é realizado por uma instância de comutador por *software* Open vSwitch; e o QFlow realiza a separação de planos, o isolamento de recursos e a garantia de QoS baseado na aplicação do *QFlow App*. Os resultados de atraso mostrados na Figura 3(a) revelam que o QFlow introduz um aumento do atraso no encaminhamento do primeiro pacote da ordem de 1 ms, quando comparado com as outras ferramentas. Esse atraso é referente ao encaminhamento do primeiro pacote de cada fluxo para o NOX, para que esse processe as informações do pacote e instale um novo fluxo no comutador OpenFlow do plano de dados. No encaminhamento dos demais pacotes, o atraso introduzido pelo QFlow é próximo ao atraso intrínseco do encaminhamento pelo Open vSwitch, já que o fluxo já está instalado no plano de dados. Isso mostra que as ações de troca de endereços MAC e direcionamento de pacotes a filas não interferem de forma significativa no atraso de processamento dos pacotes. No Xen, o atraso de encaminhamento dos pacotes é superior aos demais, pois cada pacote passa duas vezes pelo plano de dados, uma na chegada ao nó físico, quando é redirecionado ao plano de controle, e outra quando deixa o plano de controle, volta ao plano de dados e só então é transmitido no enlace de saída.



**Figura 3. Comparação dos atrasos introduzidos pelo encaminhamento de pacotes de um fluxo.**

O segundo experimento avalia o comportamento dos três sistemas, QFlow, Open vSwitch e Xen, para a diferenciação de pacotes, que é uma forma de se avaliar o isolamento entre os diferentes roteadores virtuais e também a oferta de qualidade de serviço. A plataforma Xen não é capaz de prover diferenciação de serviço em redes virtuais nativamente porque não provê qualquer tipo de controle de banda. No cenário com Open vSwitch, para realizar diferenciação de pacotes foi usada a mesma aplicação de direcio-

namento de pacotes a filas usada no QFlow, porém não é usado o controlador de banda do QFlow. Esse experimento consiste em criar três redes virtuais, Rede 1, Rede 2 e Rede 3, cada uma com, respectivamente, limite mínimo garantido de 10 Mb/s, 20 Mb/s e 30 Mb/s de banda. Para a realização do experimento, cada rede virtual é definida como um fluxo entre o gerador de tráfego e o receptor. O gerador de cada uma das três redes transmite, a uma taxa constante, 60 Mb/s de tráfego UDP de pacotes de 1472 B<sup>3</sup> para cada rede virtual. A banda total requerida pelas três redes virtuais é então de 180 Mb/s e, portanto, superior a capacidade dos enlaces de 100 Mb/s. A Figura 4 mostra como os limites são fornecidos em cada sistema. Nos três casos percebe-se a saturação dos enlaces de 100 Mb/s. No Xen, os limites mínimos não são respeitados. No Open vSwitch, cada rede virtual é direcionada a uma fila e, portanto, tem o limite mínimo da fila garantido. Contudo, o restante da banda não atribuída do enlace é igualmente dividido entre os fluxos, não sendo possível definir uma relação de prioridade entre eles. Já no QFlow, a redistribuição dos recursos ociosos é realizada de forma a privilegiar as redes virtuais com maior limite mínimo, ou seja, privilegiar a de maior prioridade.



**Figura 4. Distribuição da capacidade disponível no enlace entre redes virtuais.**

A divisão dos recursos ociosos do enlace deve respeitar a prioridade de cada rede virtual. Para avaliar o quanto o sistema QFlow respeita a prioridade de cada rede virtual, foi definida a métrica

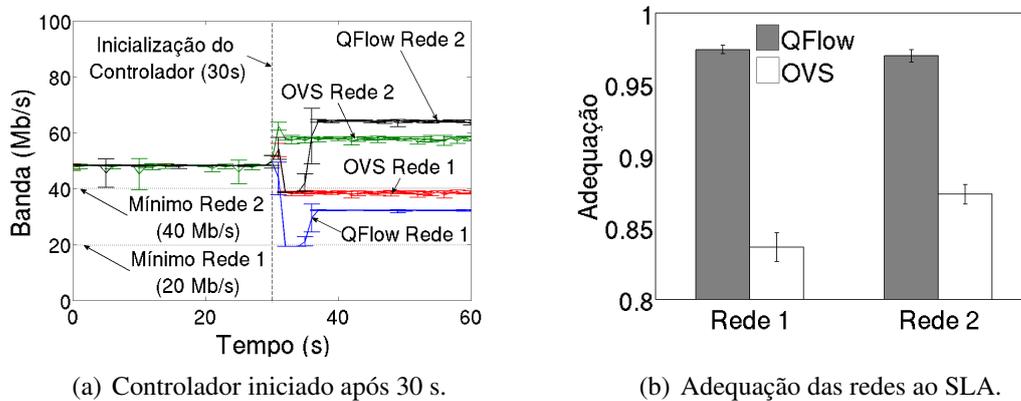
$$adequacao = 1 - \left| 1 - \frac{Banda\ Obtida}{Banda\ Esperada} \right|, \quad (2)$$

que calcula o percentual de acerto da redistribuição de recursos de acordo com a prioridade de cada rede. A métrica determina o quão próximo a banda destinada a uma rede virtual está ao seu valor ideal estimado.

O terceiro experimento verifica a eficácia do controlador de QoS do QFlow, no que se refere a uma redistribuição diferenciada e proporcional de recursos, e compara os resultados com a distribuição de recursos ociosos do Open vSwitch. Para tanto, configurou-se o Open vSwitch para direcionar os fluxos de cada rede virtual para uma determinada fila e configurou-se a banda mínima garantida de cada fila de acordo com a definição da rede e o valor máximo, por questão de justiça na comparação com o QFlow, foi configurado para a banda máxima do enlace, 100 Mb/s. No QFlow, definiu-se a banda mínima reservada para cada fila e a banda máxima é estabelecida pelo controlador. A Figura 5 apresenta o experimento para dois fluxos UDP de pacotes de 1472 B para duas redes virtuais diferentes, uma com limite mínimo de 20 Mb/s e outra com limite mínimo de 40 Mb/s. A

<sup>3</sup>A carga útil dos pacotes gerados é de 1472 B, que somados aos cabeçalhos do UDP e do IP gera um tamanho total de 1500 B, que corresponde ao tamanho máximo de conteúdo de um quadro Ethernet.

dinâmica do experimento é a seguinte. O experimento é iniciado em 0 s. No início do experimento, nenhum limite de banda é definido e o enlace é compartilhado totalmente pelas duas redes. Em 30 s, é definido os limites mínimo e máximo das filas no cenário do Open vSwitch e é iniciado o controlador de QoS no cenário do QFlow. O experimento é encerrado aos 60 s.



**Figura 5. Adequação dos controlador de QoS aos SLAs, comparado ao comportamento do Open vSwitch com parâmetros de QoS.**

A Figura 5 evidencia a eficácia da redistribuição diferenciada e proporcional a qualidade de serviço especificada na abordagem QFlow. A rede com maior banda mínima possui uma fila com maior prioridade e, conseqüentemente, tem acesso a uma maior parcela de largura da banda da repartição do excedente dos recursos ociosos do enlace. Na abordagem nativa do Open vSwitch os recursos livres no enlace são igualmente divididos entre todas as filas. Já a Figura 5(b) mostra a adequação, após a definição dos limites de cada fila. A adequação do QFlow é, em torno de, 14% superior a do Open vSwitch, chegando a 97%. Assim, o controlador de QoS do QFlow mostra-se mais apto a fornecer qualidade de serviço para fluxos com prioridade do que o uso simplesmente das primitivas de QoS do OpenFlow implementadas pelo Open vSwitch.

## 6. Conclusão

Este artigo propôs o sistema QFlow, que é capaz de isolar o consumo de recursos de cada rede virtual e ainda provê Qualidade de Serviço (QoS) tanto para uma rede virtual, como para fluxos, ou conjunto de fluxos, de uma mesma rede virtual. O sistema baseia-se em uma arquitetura híbrida de redes virtuais em que o plano de controle executa em uma máquina virtual Xen e o plano de dados é realizado em uma matriz de comutação de pacotes que implementa a interface de programação de aplicação OpenFlow. A ideia chave do sistema QFlow é atribuir os fluxos de uma rede virtual a uma fila no plano de dados e realizar o controle de banda sobre as filas no plano de dados. Um protótipo do sistema foi implementado e avaliado. Os resultados demonstram que o mapeamento dos pacotes nas filas de encaminhamento adequadas a cada rede virtual e aos seus requisitos de QoS introduz um atraso de aproximadamente 1 ms somente no primeiro pacote de cada fluxo. Contudo, esse atraso não afeta os demais pacotes do fluxo. O controlador de recursos do QFlow alcançou a utilização máxima dos enlaces aliada a uma eficiência de distribuição proporcional de recursos de aproximadamente 97%, o que é 14% superior à eficiência do Open vSwitch.

Como trabalhos futuros, pretende-se desenvolver algoritmos de controle de admissão de novas redes virtuais em nós QFlow, assim como desenvolver algoritmos de otimização de alocação de redes virtuais sobre a infraestrutura da rede física.

## 7. Referências

- [Carapinha and Jiménez 2009] Carapinha, J. and Jiménez, J. (2009). Network virtualization - a view from the bottom. In *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, pages 73–80, New York, NY, USA. ACM.
- [Egi et al. 2008] Egi, N., Greenhalgh, A., Handley, M., Hoerd, M., Huici, F., and Mathy, L. (2008). Towards high performance virtual routers on commodity hardware. In *Proceedings of the 2008 ACM CoNEXT Conference*, pages 1–12. ACM.
- [Feamster et al. 2007] Feamster, N., Gao, L., and Rexford, J. (2007). How to lease the Internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64.
- [Fernandes and Duarte 2010] Fernandes, N. and Duarte, O. (2010). XNetMon: Uma arquitetura com segurança para redes virtuais. *Anais do X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 339–352.
- [Fernandes et al. 2010] Fernandes, N., Moreira, M., Moraes, I., Ferraz, L., Couto, R., Carvalho, H., Campista, M., Costa, L., and Duarte, O. (2010). Virtual networks: Isolation, performance, and trends. *Annals of Telecommunications*, pages 1–17.
- [Fernandes and Duarte 2011] Fernandes, N. C. and Duarte, O. C. M. B. (2011). Provendo isolamento e qualidade de serviço em redes virtuais. In *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011*.
- [Gude et al. 2008] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. (2008). NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110.
- [Handley et al. 2003] Handley, M., Hodson, O., and Kohler, E. (2003). XORP: An open platform for network research. *ACM SIGCOMM Computer Communication Review*, 33(1):53–57.
- [Hao et al. 2009] Hao, F., Lakshman, T., Mukherjee, S., and Song, H. (2009). Enhancing dynamic cloud-based services using network virtualization. In *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pages 37–44. ACM.
- [Houidi et al. 2010] Houidi, I., Louati, W., Zeghlache, D., Papadimitriou, P., and Mathy, L. (2010). Adaptive virtual network provisioning. In *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, pages 41–48. ACM.
- [Keller and Rexford 2010] Keller, E. and Rexford, J. (2010). The platform as a service model for networking. In *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, pages 4–9. USENIX Association.
- [Kim et al. 2010] Kim, W., Sharma, P., Lee, J., Banerjee, S., Tourrilhes, J., Lee, S., and Yalagandula, P. (2010). Automated and scalable QoS control for network convergence. *Proc. INM/WREN*.
- [Mattos et al. 2011a] Mattos, D., Fernandes, N. C., Costa, V., Cardoso, L., Campista, M. E. M., Costa, L. H. M. K., and Duarte, O. C. M. B. (2011a). OMNI: OpenFlow MaNagement infrastructure. In *2011 International Conference on the Network of the Future (NoF'11)*, pages 52–56, Paris, France.
- [Mattos et al. 2011b] Mattos, D., Fernandes, N. C., and Duarte, O. C. M. B. (2011b). XenFlow: Um sistema de processamento de fluxos robusto e eficiente para migração em redes virtuais. In *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011*.
- [McIlory and Sventek 2006] McIlory, R. and Sventek, J. (2006). Resource virtualisation of network routers. In *High Performance Switching and Routing, 2006 Workshop on*, pages 6–pp. IEEE.
- [McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.
- [Moreira et al. 2009] Moreira, M., Fernandes, N., Costa, L., and Duarte, O. (2009). Internet do futuro: Um novo horizonte. *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC 2009*, pages 1–59.
- [Nascimento et al. 2011] Nascimento, M. R., Rothenberg, C. E., Denicol, R. R., Salvador, M. R., and Magalhães, M. F. (2011). RouteFlow: Roteamento commodity sobre redes programáveis. In *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011*.
- [Pfaff et al. 2009] Pfaff, B., Pettit, J., Koponen, T., Amidon, K., Casado, M., and Shenker, S. (2009). Extending networking into the virtualization layer. *Proc. HotNets*.
- [Sherwood et al. 2009] Sherwood, R., Gibb, G., Yap, K., Appenzeller, G., Casado, M., McKeown, N., and Parulkar, G. (2009). Flowvisor: A network virtualization layer. Technical report, Tech. Rep. OPENFLOW-TR-2009-01, OpenFlow Consortium.
- [Wang et al. 2011] Wang, R., Butnariu, D., and Rexford, J. (2011). Openflow-based server load balancing gone wild. In *Proceedings of the 11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services*, pages 12–12. USENIX Association.