Uma plataforma de roteamento como serviço baseada em redes definidas por *software*

Carlos Corrêa[1], Sidney Lucena[1], Christian Rothenberg[2], Marcos Salvador[2]

[1] Universidade Federal do Estado do Rio de Janeiro (UNIRIO) –
Rio de Janeiro - RJ

{carlos.correa,sidney}@uniriotec.br

[2]Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD) –
Campinas - SP

{esteve,marcosrs}@cpqd.com.br

Abstract. *This work proposes that the operation of an Internet AS should be oriented by a high-level description of its business policy. To materialize this view, we introduce a routing as a service platform based on RouteFlow that allows for the virtualization of IP routing operations in OpenFlow networks. Over this platform, a novel routing service was built where a single process executing BGP is used to define, in a centralized fashion, the external routing of an AS, and yet without the need to run an IGP for internal routing. The results obtained suggest that the platform serves as a tool for the execution of an AS' business policy, under the perspective of low administrative overhead.*

Resumo. *Este trabalho propõe que a operação de um AS seja orientada por uma descrição de alto nível de sua política de negócio. Para concretizar essa visão, uma plataforma de roteamento como serviço foi construída a partir da arquitetura RouteFlow, que permite a virtualização de operações de roteamento IP em redes OpenFlow. Sobre essa plataforma, um serviço de roteamento foi construído onde um único processo rodando o protocolo BGP é usado para definir, de forma centralizada, o roteamento externo de um AS, sem ainda precisar de um protocolo IGP para o roteamento interno. Os resultados obtidos sugerem que a plataforma serve como um instrumento para a realização da política de negócio de um AS, sob uma perspectiva de baixa sobrecarga administrativa.*

1. Introdução

A virtualização, de uma perspectiva de negócios, cede aos provedores de serviços baseados na Internet a chance de concentrar-se na operação de seu *serviço*, ao invés de na operação de seus *componentes*. Também tornam-se possíveis novas abstrações de gerenciamento que encapsulam o mapeamento entre os contextos virtual e físico. Assim, a infraestrutura subjacente comporta-se menos como um conjunto de reservas individuais de capacidade, aproximando-se mais de uma reserva única de insumos computacionais, consumida conforme as instâncias de sistemas definidas pelo operador. Tais conceitos fundamentam diferentes contribuições à operação de roteadores, com foco na racionalização do uso de recursos [Egi et al. 2008] e centralização do plano de controle [Nascimento et al. 2011], dentre outras.

Em [Keller and Rexford 2010], no entanto, observa-se que apesar das inovações baseadas em plataformas virtuais oferecerem uma visão abstrata dos recursos disponíveis, ainda são gerenciadas como redes tradicionais. Cabe ao operador estabelecer a conectividade entre seus roteadores virtuais, configurar individualmente os protocolos a serem executados, como o BGP e, possivelmente, um IGP, além de gerenciar aspectos de sua capacidade e dos canais de comunicação correspondentes. Isto significa que apenas parte do esforço de gerenciamento dos equipamentos de um provedor de telecomunicações (ex.: unificação e automação de processos, auto-configuração) possa ser reduzido ou eliminado pelo uso de tecnologias de virtualização. E mesmo que todo o plano de controle de sua rede venha a ser consolidado e centralizado, é necessário estabelecer comunicação entre diferentes processos de roteamento. Tais processos precisam estabelecer adjacências, hierarquias e trocar mensagens de controle, tal como se estivessem operando sobre dispositivos distintos. Nesse contexto, a apresentação do plano de controle de roteamento em uma abstração de mais alto nível ainda é um problema em aberto. Mais ainda, as decisões de cada elemento roteador são tomadas a partir de suas visões particulares da rede, e de uma política global de encaminhamento, aplicada uniformemente a todos os dispositivos. Assim, traduzir os requisitos de negócio de um provedor de telecomunicações (a operação de seu *serviço*) em atividades de configuração de todos esses componentes é um desafio da área de roteamento Internet [Zhang-Shen et al. 2008].

O presente trabalho aborda ambos os problemas a partir de intervenções à arquitetura virtual de roteamento IP RouteFlow [Nascimento et al. 2011], que promove a centralização da execução de roteadores virtuais no elemento controlador de uma rede baseada em OpenFlow. Propõe-se a conversão do RouteFlow em uma *plataforma de roteamento como serviço*. Tal solução simplifica o gerenciamento de topologias virtuais de roteamento, ao dispensar a necessidade de que seu desenho mantenha uma relação 1:1 para com a conectividade física existente. Também permite a tomada de decisões de roteamento complementares àquelas realizadas pelos roteadores, agora sob uma visão global da rede, e através de componentes de *software* (serviços) especificados por seu operador.

Assim, o sentido de "serviço" aproxima-se daquele empregado no desenho de arquiteturas empresariais digitais [Papazoglou et al. 2007]: pretende-se apoiar elementos que desempenhem partes independentes de um processo de negócio, opacos entre si, mas que por suas naturezas complementares alcançam um resultado conjunto (o "roteamento" de tráfego). Finalmente, para que um operador possa implementar serviços arbitrários e descrever de forma única o comportamento esperado de uma arquitetura virtual de roteamento IP, espera-se prover uma "plataforma", como um conjunto de interfaces e abstrações de alto nível compartilhadas pelos serviços.

O texto das próximas seções está assim organizado: a Seção 2 apresenta uma revisão da operação do protocolo BGP e da plataforma OpenFlow, além das arquiteturas virtuais de roteamento do estado-da-arte; na Seção 3 a plataforma almejada é descrita, bem como seus detalhes de implementação; a Seção 4 apresenta um exemplo de aplicação escrita para a plataforma, avaliada através de um estudo de caso; os resultados obtidos são discutidos na Seção 5, além de serem comparativamente analisados em relação aos de outras soluções do estado-da-arte; por fim, a Seção 6 é reservada para apresentação das conclusões e trabalhos futuros.

2. Background

2.1. O BGP e os desafios do roteamento Internet

O protocolo BGP (do inglês *Border Gateway Protocol*) é o elemento responsável por divulgar a alcançabilidade entre diferentes redes na Internet. A operação BGP assume que cada rede (também chamada "sistema autônomo", ou AS, na sigla em inglês) possa ser tratada como uma unidade atômica. Porém, isso nem sempre é verdadeiro, porque cada uma delas pode ser composta por múltiplos roteadores, que tomam decisões geralmente independentes entre si [Zhang-Shen et al. 2008].

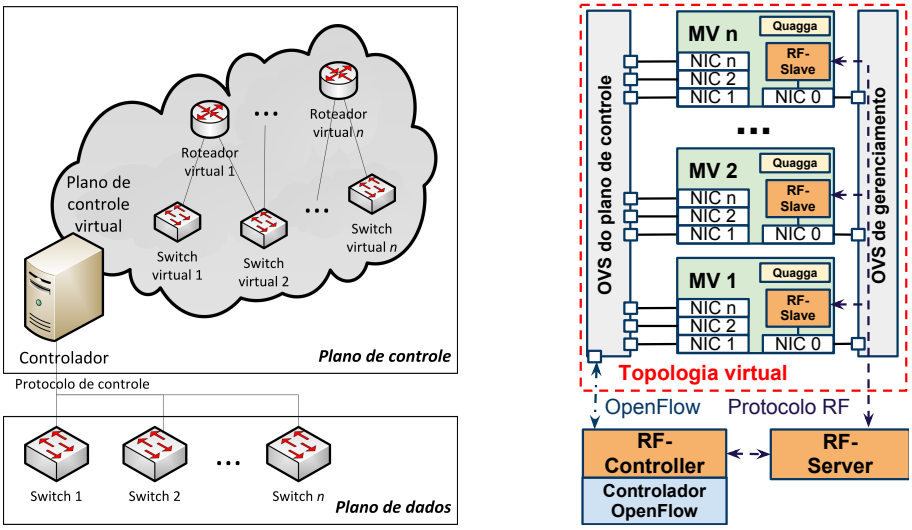
Tal premissa implica em um alto custo de gerenciamento. Cada roteador executa uma instância BGP que possui independência nas decisões de encaminhamento, mas devem guardar coerência entre si. A despeito disso, os provedores de serviços Internet precisam que essas decisões estejam de acordo com seus contratos de transporte de dados, ou seja, suas políticas de negócio [Zhang-Shen et al. 2008]. Até o momento, a solução mais comum para esse impasse envolve ajustar os parâmetros do BGP em cada roteador do AS, na expectativa de que produzam, coletivamente, um comportamento consistente com a política de negócio vigente. Porém, isso nem sempre é possível, porque além desses parâmetros, características da topologia da rede ou a execução de outros protocolos podem interferir no roteamento resultante [Teixeira et al. 2004], [Zhang-Shen et al. 2008] e [Park et al. 2010].

Uma vez que a operação BGP compreende apenas o aspecto inter-redes, a execução de um protocolo de roteamento IGP (do inglês *Interior Gateway Protocol*), capaz de tratar a alcançabilidade entre os nós que compõem uma rede, faz-se necessária. Assim, o processamento de IGPs também afeta as decisões de encaminhamento, inclusive aquelas que dirigem tráfego para fora do AS. Isto implica na necessidade de se manter mais um artefato de configuração ativo e consistente com a política de negócio vigente. O exemplo mais claro dessa relação entre IGP e BGP é conhecido como roteamento *hot-potato*. Trata-se de uma regra de desempate usada pelo BGP quando o mesmo conhece mais de um caminho de mesmo custo para uma dada rede destino, já considerando outros critérios mais prioritários para a seleção de rotas. Nesse caso, o BGP opta pelo caminho cujo ponto de saída do AS tem menor distância IGP. Em [Teixeira et al. 2004] é mostrado como instabilidades no roteamento IGP, devido a oscilações de enlaces, por exemplo, podem afetar todo um caminho inter-ASes até a rede de destino.

2.2. A arquitetura OpenFlow

A arquitetura de redes definidas por *software* OpenFlow prevê a generalização da função desempenhada pelos *switches* de uma rede, renomeando-os *datapaths*. Tal generalização faz com que cada *datapath*, ao receber um pacote de dados para encaminhamento, baseie sua decisão em entradas de uma tabela de fluxos. Essa tabela possui colunas que descrevem possíveis características das unidades de tráfego a serem encaminhadas, fazendo uso de campos dos cabeçalhos das diversas camadas de comunicação. Um campo final descreve as ações a serem realizadas para as entradas que possuem um mesmo conjunto de características.

O protocolo OpenFlow é usado para transmitir a um *datapath* instruções quanto a inserir, modificar ou remover entradas de sua tabela de fluxos. Isto porque sua definição não acontece nos próprios dispositivos, mas originam-se em um novo elemento da rede,



(a) Modelo conceitual. (b) Arquitetura RouteFlow.

Figura 1. Componentes das arquiteturas virtuais de roteamento IP.

o controlador OpenFlow. O controlador é um servidor que implementa uma lógica de decisão a partir da qual são derivadas as instruções a serem transmitidas.

Os dados de entrada para a lógica decisória dos controladores são os próprios cabeçalhos dos pacotes recebidos por um *datapath*. Assim, tão logo esse tipo de elemento tenha uma unidade de tráfego a tratar, extrai os dados de seu cabeçalho e os remete, como uma pergunta, ao controlador. Isso dispara o processo de derivação de entradas que permitirá ao controlador instruir que tratamento o *datapath* deve aplicar ao pacote.

Ao serem incorporadas à tabela de fluxos, as instruções de encaminhamento recebidas podem ser mantidas por um tempo arbitrário na memória de um *datapath*. Porquanto uma entrada da tabela seja mantida, todos os pacotes recebidos por um *datapath* que tenham características coincidentes com as especificadas por ela receberão automaticamente o mesmo tratamento.

Uma vez que a lógica de decisão dos elementos controladores pode ser implementada como *software* e basear-se em combinações arbitrárias de campos de cabeçalhos, a arquitetura OpenFlow pode facilitar a inovação, experimentação e operacionalização de idéias no âmbito das redes de computadores. Para reduzir a lacuna entre o tratamento de informações de baixo nível envolvidas no protocolo OpenFlow e a construção de aplicações que interajam com esta arquitetura, *softwares* controladores de uso geral também estão disponíveis [Gude et al. 2008] [of Stanford 2010]. Essas ferramentas oferecem interfaces para a implementação de abstrações de mais alto nível que aquelas cabíveis no nível do protocolo.

2.3. Arquiteturas virtuais de roteamento IP

Nestas arquiteturas, o plano de controle é representado por um sistema controlador, que possui em sua memória uma representação da topologia da rede controlada. Isso pode ser visto na Figura 1(a), que mostra uma rede atendida por uma plataforma virtual de roteamento IP. Os n comutadores da rede estão interligados em série, tornando-a plena-

mente conexa no nível de enlaces. Cada comutador físico é representado na memória do controlador por uma instância de um mecanismo de conectividade virtual.

Em uma arquitetura tradicional, para permitir a comunicação entre clientes dessa rede, configurados em diferentes domínios de *broadcast*, seria preciso empregar um dispositivo com função de roteamento (um roteador, por exemplo). No cenário apresentado, porém, máquinas virtuais (MVs) executando processos de roteamento são suficientes para que esta função seja desempenhada. O controlador consulta as bases de informações de roteamento dos processos do plano de controle virtual. Assim, é possível determinar por quais enlaces os dados devem ser transmitidos para chegar a seu destino, bem como comandar os elementos do plano de dados de acordo.

A arquitetura virtual utilizada no contexto deste trabalho é a RouteFlow [Nascimento et al. 2011], de código aberto. Sua estrutura pode ser sumarizada em 3 componentes básicos: o RF-Slave, o RF-Server e o RF-Controller. O RF-Slave é o módulo que se acopla a cada MV executando processos de roteamento, para coletar as entradas das tabelas de encaminhamento (FIBs, do inglês *Forwarding Information Base*) e exportá-las para o RF-Server. O RF-Server é responsável pela lógica de controle do RouteFlow, mapeando datapaths OpenFlow em MVs, recebendo eventos, comandando a inserção de entradas de fluxos e mantendo o estado da rede. O RF-Controller é o módulo responsável pela interface de comunicação entre RF-Server e o controlador OpenFlow, como o NOX. A Figura 1(b) ilustra essa arquitetura.

3. Construção de uma plataforma de roteamento como serviço

Nas arquiteturas virtuais de roteamento IP, como a RouteFlow, decisões triviais de um protocolo como o BGP são tomadas por processos individuais, virtualmente conectados. Em seguida seus resultados são coletados e submetidos a um processo controlador. Tal processo é responsável por apenas traduzir decisões de encaminhamento em instruções do plano de dados.

O objetivo geral deste trabalho é estabelecer uma plataforma de roteamento como serviço, que seja capaz de extrair decisões de roteamento de um plano de controle virtual e submetê-las a lógicas de encaminhamento arbitrárias. Tais unidades lógicas podem modificar ou complementar as decisões tomadas, oferecendo uma interface mais flexível para a realização da política de negócio de um provedor Internet. Cada possível unidade lógica aplicada à plataforma é aqui denominada um *serviço* de roteamento.

Espera-se que a arquitetura RouteFlow, por sua característica de "pós-processamento" centralizado, possa incorporar a lógica necessária para atender a uma política de negócio arbitrária, dependendo apenas da disponibilidade de meios para que o operador da rede a descreva – uma gramática de configuração. Tal gramática também deve permitir a descrição da topologia física da rede a ser controlada, o que vem instrumentalizar o gerenciamento automatizado da conectividade disponível.

3.1. Premissas para a gramática de configuração

A concepção da gramática é apoiada por proposições menos gerais, elencadas a seguir.

Proposição 1 (Expressividade)**.** *Uma política de roteamento deve ser integralmente descrita nos termos da gramática para que possa ser completamente atendida.*

Toda lógica de processamento complementar àquela existente nos protocolos de roteamento suportados pelo RouteFlow deve ser implementada como parte do componente RF-Server. Cada unidade de lógica deve ser disponibilizada ao operador da rede como um elemento específico da sintaxe da gramática de configuração. Caso a execução da funcionalidade incorporada dependa de parâmetros de entrada, estes também deverão poder ser expressos como elementos da mesma sintaxe.

Proposição 2 (Satisfabilidade)**.** *A política de roteamento descrita através da gramática de configuração deverá ser sempre satisfazível pela topologia subjacente.*

É responsabilidade do operador especificar e prover a infraestrutura capaz de atender à descrição de sua política de negócio. Não obstante, assim como o nível de abstração oferecido por uma plataforma de virtualização permite uma visão consolidada dos recursos disponíveis, mas não pode utilizá-los além de seus limites, é desejável que a plataforma vislumbrada rejeite uma descrição de política que obviamente exceda a capacidade da infraestrutura existente.

Proposição 3 (Tempestividade)**.** *A seleção de rotas que permite implementar a política de negócio vigente deve ser sempre recalculada, bem como seus resultados transpostos para o plano de dados, em resposta a uma nova decisão de roteamento (convergência).*

Tal como em uma rede tradicional, as convergências dos protocolos de roteamento acontecem em resposta a toda modificação de topologia, seja motivada por desconexões, reconexões ou o estabelecimento de novas enlaces, por exemplo. Na arquitetura RouteFlow a identificação e resposta a novas convergências baseia-se em uma coleta periódica das tabelas de encaminhamento de cada roteador virtual, o que pode resultar em períodos com política de roteamento desatualizada. Dessa forma, para garantir que a política vigente sempre reflita as informações produzidas pela última convergência ocorrida na rede controlada, o RouteFlow deverá incorporar um mecanismo de resposta a eventos de roteamento, que atualizará a lógica de encaminhamento de acordo com as decisões de roteamento obtidas.

3.2. Base de informações da topologia física

Assim como em [Caesar et al. 2010], a plataforma de roteamento como um serviço proposta vislumbra que o roteamento de um AS deve ser executado sob uma supervisão centralizada. Assim sendo, coloca-se aqui o requisito de que o operador descreva a topologia física da rede a ser controlada pela plataforma. Tal descrição vem permitir que sejam implementados serviços que se baseiem diretamente nas conexões entre dispositivos para determinar a alcançabilidade entre eles, possivelmente minimizando ou até eliminando o uso de um IGP [Caesar et al. 2010]. Para efeito de ilustração, suponha-se o seguinte trecho de código:

```
dpid 0x1 ports 2
dpid 0x2 ports 3
dpid 0x1 port 2 trunk dpid 0x2 port 1
```

Nesse caso, duas primeiras linhas do exemplo informam que a infraestrutura a ser controlada conta com dois *datapaths*, de identificadores “0x1” e “0x2”. Eles possuem duas e três portas, respectivamente. A terceira e última linha informa que a porta de número 2 no *datapath* “0x1” está conectada à porta 1 do *datapath* “0x2”. Um serviço hipotético de roteamento poderia utilizar o conhecimento sobre o plano de dados para instruir o *datapath* “0x2” a transmitir pela porta 1 o tráfego destinado a prefixos alcançáveis por meio do *datapath* “0x1”, sem o uso de qualquer protocolo.

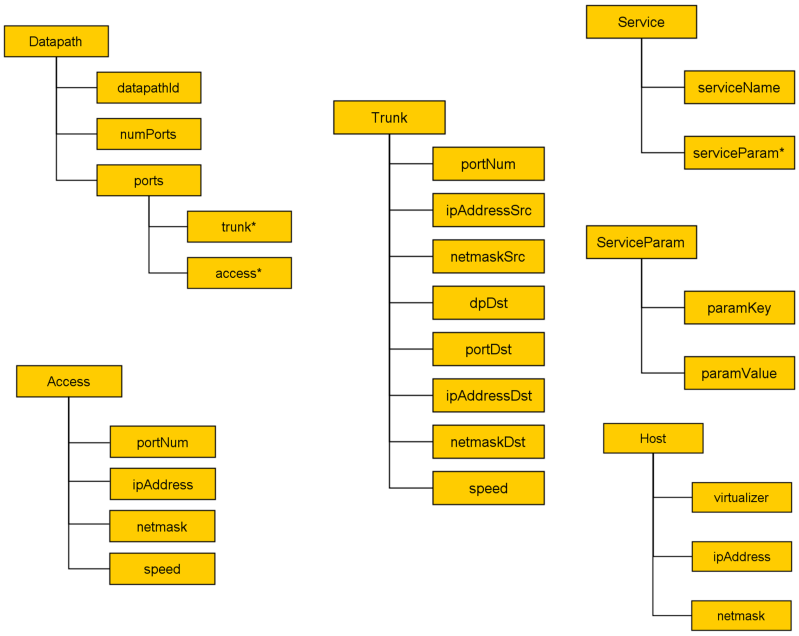


Figura 2. Modelos de dados para os elementos do plano de controle RouteFlow.

3.3. Representação de elementos

De acordo com os objetivos propostos, do ponto de vista do plano de dados a gramática limita-se à representação de *datapaths*, suas portas de comunicação e suas funções, definidas como duas para efeito desta pesquisa: a ligação entre dois *datapaths* e a conexão do *datapath* a outros tipos de dispositivo. Na perspectiva de controle é possível descrever os sistemas anfitriões que executam a topologia virtual e os serviços que se deseja incorporar ao roteamento, bem como seus parâmetros.

A Figura 2 apresenta o conjunto de elementos considerados para a construção da gramática, com seus atributos. A título de exemplo, segue a descrição de alguns deles:

Trunk. Apresenta o modelo de representação de uma porta de comunicação usada para interligar um *datapath* a outros aparelhos deste tipo, dentro do mesmo AS. A porta em questão é identificada pelo atributo "portNum" (número de porta). A outra extremidade do enlace, por sua vez, está designada nos campos "dpDst" e "portDst", que indicam respectivamente o *datapath* e a porta remotos. Também estão previstos os atributos de endereço IP e máscara de rede para essas conexões, além de um campo para especificação de velocidade de transmissão.

Access. Portas de comunicação utilizadas para conexão entre *datapaths* e outros tipos de dispositivos são consideradas portas de acesso. Possui atributos análogos aos do elemento Trunk.

Service e ServiceParam. O elemento "Service" incorpora as informações pertinentes a um serviço de roteamento da plataforma RouteFlow. Sua função é identificar cada serviço por seu respectivo nome ("serviceName") e estabelecer uma relação $1 : n$ com os parâmetros de operação especificados para o mesmo. Tais parâmetros estão incorporados ao elemento "ServiceParam" que oferece uma estrutura de chaves ("ParamKey") e valores ("ParamValue") para informação de um serviço a respeito de seu comportamento

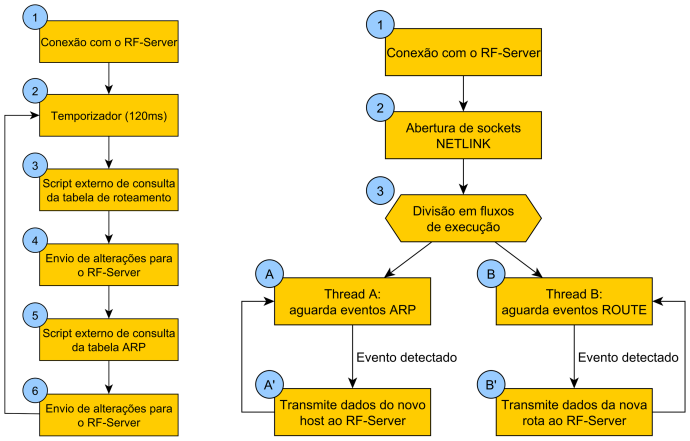


Figura 3. Diferentes modelos de fluxos de execução para a lógica do RF-Slave.

esperado.

3.4. Tempestividade

Conforme a proposição 3 da Seção 3.1, era necessário incorporar ao RF-Slave mecanismos de coleta de dados que reagissem a eventos de convergência ocorridos no plano de controle. Isso permitiu identificar três passos necessários à implementação da funcionalidade planejada:

1. Incorporar a coleta de dados da FIB ao próprio RF-Slave, eliminando o uso de *scripts* externos;
2. Condicionar o processo de coleta à ocorrência de eventos na FIB;
3. Garantir que múltiplos fluxos de execução possam ser utilizados, de maneira que diferentes eventos (ou *tipos* de eventos) possam ser tratados paralelamente.

O sistema Linux, plataforma-alvo do RouteFlow, conta com uma estrutura interna que simula uma comunicação de rede (*socket*) entre o núcleo do SO e um processo de espaço de usuário, chamada NETLINK. Em complemento à NETLINK, está disponível um mecanismo de coleta de informações da FIB do sistema operacional, chamado RTNETLINK [Udugama 2006]. Todas as alterações realizadas na FIB são anunciadas via *sockets* RTNETLINK e transmitidos para grupos *multicast*, onde cada grupo corresponde a uma categoria, como “tabela de roteamento” ou “tabela ARP”. Dessa forma, o RTNETLINK apresenta-se como uma opção de coleta de dados tempestiva e independente da suíte de roteamento em uso.

A Figura 3 permite comparar a operação do RF-Slave antes e depois da implementação RTNETLINK. Seu fluxo de execução, originalmente monolítico, foi dividido em duas *threads*. Uma *thread* do agente passou a estar inscrita no grupo *multicast* de anúncios sobre a tabela de roteamento, enquanto outra trata exclusivamente das atualizações da tabela de vizinhança (ARP). Quando uma mensagem é recebida, os dados relevantes de cada tipo de anúncio são extraídos e enviados ao RF-Server para instrução do plano de dados.

4. Aplicação

A plataforma RouteFlow deve proporcionar uma infraestrutura de suporte a aplicações de roteamento, tanto quanto o sistema operacional de rede NOX proporciona uma fundação

para a execução de controladores OpenFlow. Para validar essa proposição, foi desenvolvido um serviço de roteamento para a plataforma que permite mapear um conjunto de *datapaths*, funcionalmente equivalentes a roteadores no contexto RouteFlow, para uma única MV do plano de controle. Assim, ao invés de se gerenciar de forma distribuída n ou mais sessões BGP estabelecidas entre um AS e seus n pares, todas elas passam a ser terminadas em um único roteador virtual.

Não foi possível identificar uma categoria específica na literatura que permita enquadrar esta ferramenta, razão considerada suficiente para nomeá-la "RFAgg", cunhando-se também a expressão "aplicação de roteamento agregado".

Esta aplicação depende da característica de *expressividade*. É preciso informar a ela, por intermédio da gramática especificada, quais dispositivos farão parte da operação agregada. Ela também implementa o teste de *satisfabilidade* previsto, de forma a avaliar a infraestrutura física disponível, descrita também por meio da gramática de configuração. Sua operação é compatível com os *instrumentos de virtualização* existentes na plataforma e, por fim, sua lógica de operação é complementar àquela de um protocolo de roteamento, na medida em que decisões baseadas em uma visão global da conectividade são tomadas. Isso se opõe à execução distribuída para a qual estes mesmos protocolos foram concebidos.

O requisito para a execução da função proposta é a existência de conectividade *full meshed* entre os *datapaths* que se deseja agregar. Sabe-se que nem sempre é possível implementar uma topologia do tipo em ASes do mundo real, porém, recursos existentes no arcabouço OpenFlow sugerem que esse modelo de operação possa ser extrapolado para redes onde a configuração física não é, de fato, uma malha completa. Por exemplo, é possível o uso de túneis MPLS para a criação de canais lógicos de comunicação entre elementos de um AS [Verkaik et al. 2007].

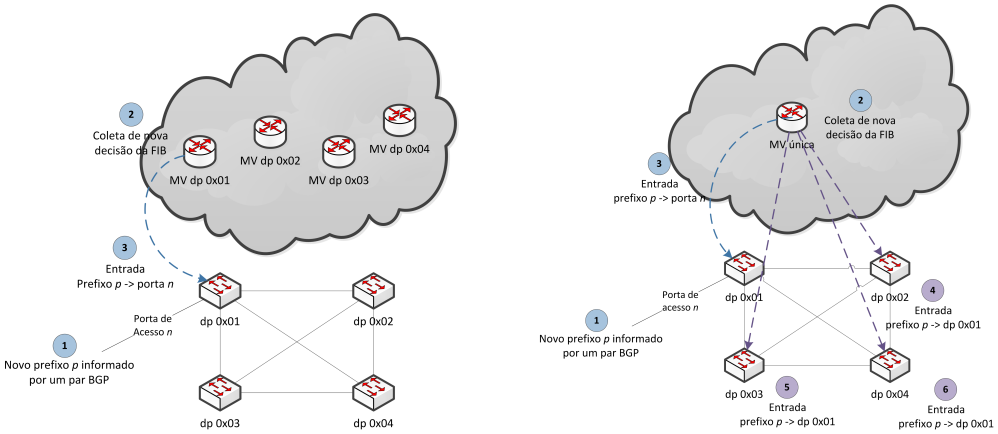
4.1. Modelo de operação

O serviço desenvolvido considera apenas dois parâmetros para sua execução: "active" e "datapaths". O primeiro ativa ou desativa a funcionalidade provida pela aplicação. O segundo é utilizado para informar a lista de *datapaths* que compõem o cenário de agregação.

A partir da interpretação da lista de dispositivos, torna-se possível consultar elementos do tipo "datapath", mantidos pela lógica principal da plataforma e construídos com base na mesma gramática de configuração. É possível então explorar exaustivamente a conectividade entre *datapaths* descrita por esses elementos, obtendo-se assim a matriz de adjacências correspondente à infraestrutura da rede. Constituída a matriz de adjacências, torna-se trivial o teste de satisfabilidade do plano de dados.

A ação crucial desempenhada por uma arquitetura virtual de roteamento IP envolve a instalação de entradas de encaminhamento no plano de dados, que sintetizam o conhecimento da alcançabilidade entre pares de endereços de origem e destino. Portanto, para o caso de agregação, após a instalação de uma entrada D em um *datapath*, referente a uma rota conhecida pela MV agregadora, é necessário um último passo: instalar uma entrada correspondente à primeira em cada um dos outros *datapaths* do domínio.

Na Figura 4 dois processos de instalação de entradas, com e sem a execução da aplicação de roteamento agregado, estão ilustrados. No caso da Figura 4(a), que mos-



(a) Instalação regular. (b) Instalação com roteamento agregado.

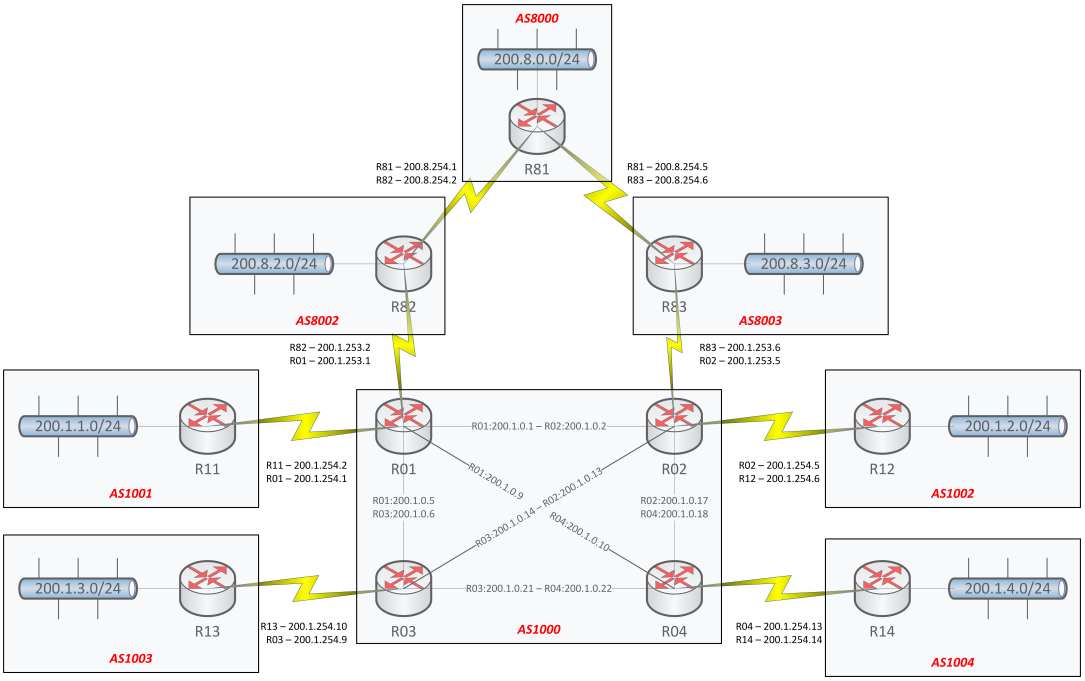
Figura 4. Diferentes processos de instalação de entradas para *datapaths*.

tra o comportamento regular da plataforma, a descoberta do prefixo p, através da porta n ligada ao *datapath* 0x01, motiva a inclusão de uma nova rota na FIB da MV para a qual o dispositivo "0x01" está mapeado (por exemplo, a MV que abriga o roteador virtual R01), acarretando na inserção do respectivo fluxo naquele *datapath*. De maneira consistente, esse processo irá se repetir em cada *datapath* cujo respectivo roteador virtual receber o anúncio de p - por exemplo, através de um IGP. Com o serviço de agregação, como representado na Figura 4(b), a descoberta do prefixo p na porta n do datapath 0x01 motivará a atualização da FIB da MV agregadora, o que acarretará com que não somente este *datapath* seja instruído, mas também os demais *datapaths* sejam informados da disponibilidade de acesso a p, garantindo a consistência do roteamento em todo o domínio mesmo sem o uso de IGP.

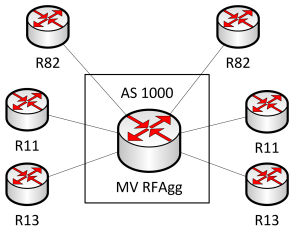
4.2. Estudo de caso

A aplicação experimental do serviço RFAgg ocorreu em uma rede inteiramente virtual, em um computador executando o Debian GNU/Linux 5.0 e as ferramentas LXC, para virtualização, e Open vSwitch (OVS), para emulação de datapaths OpenFlow. O domínio no qual essa aplicação será executada é o AS 1000, representado na Figura 5(a) juntamente com outros ASes. Apesar de hipotético, ele guarda semelhanças para com a média dos sistemas autônomos em produção na Internet. O AS conta com 6 conexões a outros ASes, quando a média mundial é de 3,37 conexões, e a brasileira, 3,54 [Alves et al. 2010]. Ele tem uma relação de fornecimento de acesso aos ASes 1001, 1002, 1003 e 1004, além de trocar trânsito com os ASes 8001 e 8002, clientes do AS 8000.

Numa operação tradicional, este arranjo importa porque demanda a manutenção de dois protocolos de roteamento distintos no AS 1000: o BGP e um IGP (como o OSPF). Por si só, tal aspecto já representa significativa sobrecarga administrativa. Ao mesmo tempo, esse cenário pode levar a problemas de *loops* de roteamento e atrasos de convergência relatados em [Park et al. 2010], decorrentes de possíveis instabilidades nos enlaces internos. Além disso, a topologia não atende às características demonstradas em [Zhang-Shen et al. 2008] como necessárias para uma operação atômica, como a conexão de cada roteador a um único tipo de AS vizinho (cliente ou parceiro de troca de tráfego). Essa condição pode limitar a escalabilidade da rede estudada.



(a) Topologia original da rede.



(b) Topologia agregada.

```
dpid 0x1 ports 5
dpid 0x2 ports 5
dpid 0x3 ports 4
dpid 0x4 ports 4

dpid 0x1 port 1 trunk dpid 0x2 port 1 speed 1000
dpid 0x1 port 2 trunk dpid 0x3 port 1 speed 1000
dpid 0x1 port 3 trunk dpid 0x4 port 1 speed 1000

dpid 0x2 port 2 trunk dpid 0x3 port 2 speed 1000
dpid 0x2 port 3 trunk dpid 0x4 port 2 speed 1000

dpid 0x3 port 3 trunk dpid 0x4 port 3 speed 1000

service rfagg active: yes
service rfagg datapath: 0x1 0x2 0x3 0x4
```

(c) Configuração RouteFlow.

```
password routeflow
enable password routeflow
!
router bgp 1000
        no synchronization
        redistribute connected
        neighbor 200.1.254.2 remote-as 1001
        neighbor 200.1.254.6 remote-as 1002
        neighbor 200.1.254.10 remote-as 1003
        neighbor 200.1.254.14 remote-as 1004
        neighbor 200.1.253.2 remote-as 8002
        neighbor 200.1.253.6 remote-as 8003
!
log file /var/log/quagga/bgpd.log
!
```

(d) Configuração “bgpd”.

Figura 5. Detalhes da operação do estudo de caso RFAgg.

Para aplicar a arquitetura RouteFlow original ao controle do AS1000 é necessário substituir os seus roteadores R01, R02, R03 e R04 por *datapaths*. Cada dispositivo tem suas portas de comunicação mapeadas para as interfaces de uma MV. Cada MV faz parte de um plano de controle em que desempenha o antigo papel dos roteadores que foram eliminados.

Tal abordagem pode ter grandes resultados sob o aspecto de desempenho. Porém, do ponto de vista de gerenciamento os ganhos são menos perceptíveis: para operar a rede do AS 1000 é necessário gerenciar sua topologia virtual, que está sujeita aos mesmos desafios de configuração da versão “tradicional” da rede. Mas, aplicando-se ao cenário o serviço RFAgg, uma nova perspectiva se configura. Essa mudança passa por um paradigma de mapeamento mais flexível entre os planos físico e virtual, representada na Figura 5(b). Nesse caso, do AS 1000, as portas de acesso da rede são mapeadas em uma relação “1:1” com seis interfaces de uma MV que representa a agregação de dispositivos.

O primeiro passo é o uso da gramática de configuração RouteFlow para descrever a topologia do AS 1000. Esta descrição encontra-se reproduzida na Figura 5(c), onde

cada *datapath* é declarado com seu respectivo número de portas. As portas de tronco são todas descritas, portas não descritas são automaticamente consideradas de acesso.

É preciso um último passo: a ativação do protocolo BGP sob a perspectiva global almejada. Na configuração única, listada na Figura 5(d), estão reunidas as declarações de todas as sessões BGP fechadas com outros ASs. Pela definição do próprio serviço RFAgg, nenhuma outra configuração de protocolo é necessária.

Durante a execução da rede foi possível verificar que as informações de alcançabilidade foram corretamente disseminadas. Também verificou-se que os roteadores operando de forma legada (R11, R12, R13, R14, R82, R83 e R81) incorporaram à sua FIB tanto os prefixos de redes diretamente conectadas quanto os de redes remotas. Também foram bem-sucedidas tentativas de comunicação a partir de todos os roteadores do cenário, rumo a todas as interfaces de rede disponíveis.

A Tabela 1 lista as rotas presentes na memória da MV de agregação, onde os *gateways* "0.0.0.0" indicam rotas diretamente aprendidas da configuração das interfaces. Como se pode ver, a conectividade plena da rede se estende a essa abstração que, estabelecendo com sucesso sessões BGP para com todos os seus pares, incorporou todas as informações de seus prefixos remotos à sua própria FIB.

Tabela 1. Rotas da MV Agg

Rede destino	*Gateway*	Máscara	Interface
MV - R13	0.0.0.0	/30	eth5
MV - R14	0.0.0.0	/30	eth6
MV - R11	0.0.0.0	/30	eth1
MV - R12	0.0.0.0	/30	eth3
MV - R82	0.0.0.0	/30	eth2
MV - R83	0.0.0.0	/30	eth4
R82 - R81	R82	/30	eth2
R83 - R81	R83	/30	eth4
AS 1004	R14	/24	eth6
AS 1001	R11	/24	eth1
AS 1002	R12	/24	eth3
AS 1003	R13	/24	eth5
AS 8002	R82	/24	eth2
AS 8003	R83	/24	eth4
AS 8000	R83	/24	eth4

Tabela 2. Tabela de fluxos do *datapath* 0x01

Núm.	Destino	Ações
1	R13	output:2
2	R12	output:1
3	R11	ARP_{dst} = MAC_{r11}, ARP_{src}=MAC_{MV}, output:4
4	R83	output:1
5	R14	output:3
6	R82	ARP_{dst} = MAC_{r82}, ARP_{src}=MAC_{MV}, output:5
7	AS 8000	output:1
8	AS 1002	output:1
9	AS 1001	ARP_{dst} = MAC_{r11}, ARP_{src}=MAC_{MV}, output:4
10	AS 1003	output:2
11	AS 1004	output:3
12	AS 8002	ARP_{dst} = MAC_{r82}, ARP_{src}=MAC_{MV}, output:5
13	AS 8003	output:1
14	Rede R83 - R81	output:1
15	Rede R82 - R81	ARP_{dst} = MAC_{r82}, ARP_{src}=MAC_{MV}, output:5

A partir da consulta às entradas de fluxos OpenFlow instaladas no plano de dados, foi possível verificar que o processo de coleta de dados da FIB, bem como sua tradução em uma lógica de encaminhamento distribuída, foi bem-sucedido. A Tabela 2 lista as entradas OpenFlow existentes no *datapath* 0x01 após a completa inicialização da topologia. Entradas respectivas foram verificadas nos três outros datapaths, mas foram omitidas por concisão.

5. Discussão

A solução RFAgg aqui apresentada reduz a operação de um AS à manutenção de uma única unidade de encaminhamento. Isso aproxima-se do conceito de roteamento atômico apresentado em [Zhang-Shen et al. 2008], sem no entanto requerer modificações ao protocolo BGP.

Parâmetros adicionais poderiam ser incorporados ao serviço RFAgg. Eles poderiam designar, para um prefixo que tivesse múltiplas rotas disponíveis, qual delas melhor atende aos interesses do AS, em complemento à comparação de seus vetores de distâncias. A escolha de "melhor rota" pelo processo BGP deve ainda ser tratada. Uma vez que apenas uma rota é incorporada à FIB da MV, devido à implementação de roteador virtual

(Quagga) usada pelo RouteFlow, não há múltiplas opções disponíveis para a aplicação RFAgg. De forma a não se alterar a maneira como o RouteFlow coleta rotas e as instala nos datapaths, uma opção seria substituir o *software* de roteamento utilizado no plano de controle por uma versão capaz de instalar na FIB múltiplas "melhores rotas" de mesmo custo. Alternativamente, poderia se optar por extrair informações de caminhos diretamente da base de informações de roteamento do Quagga, modificando-se o RF-Slave para se comportar como um módulo Quagga.

Em relação ao requisito de contar-se com uma rede em malha completa, poderia-se calcular, a partir da matriz de adjacências de uma topologia arbitrária, caminhos entre todos os pares de vértices (*datapaths*) do grafo de conectividade do AS. A transmissão de pacotes pela rede poderia ser então baseada em circuitos virtuais, através do emprego de *tags* de VLANs ou *labels* MPLS [Sharafat et al. 2011], que podem ser programaticamente definidos em um contexto SDN. O uso de túneis *IP-in-IP* entre as bordas da rede poderia ser outra opção para o encaminhamento intra-domínio. O conceito de simular caminhos de um único salto também aparece em [Caesar et al. 2010], onde sugere-se o emprego de MPLS como instrumento para a comutação de pacotes.

Por fim, é possível estabelecer uma comparação entre os resultados verificados para plataforma RouteFlow e aqueles apresentados para a plataforma RCP [Caesar et al. 2005], que também preconiza um modelo centralizado de cálculo e disseminação de rotas. A RCP opera através de processos de roteamento que são integrados às topologias BGP e IGP de um AS. Em contrapartida, para o caso específico do serviço de roteamento aqui implementado, não é necessário qualquer IGP. Isso também pode representar uma vantagem quando se consideram os problemas relacionados à atrasos de convergência discutidos na literatura, e aos quais a RCP está sujeita [Caesar et al. 2005].

Vale notar que o cenário abordado no estudo de caso considera o AS 1000 como servindo apenas trânsito para os ASes vizinhos. Isso significa que os roteadores desse AS não possuem redes clientes a serem anunciadas internamente e externamente. Entretanto a solução proposta também comporta este outro cenário.

6. Conclusão

Este trabalho apresenta uma solução para o estabelecimento de uma plataforma de roteamento como serviço, que permite a livre implementação da política de negócio de um AS que faça uso de uma infraestrutura de rede baseada em *datapaths* OpenFlow. Tal plataforma é estabelecida com base na arquitetura RouteFlow, que por sua vez fornece uma solução de roteamento virtual alinhada com a arquitetura de roteamento já existente na Internet. Para tal, foi especificada uma gramática de configuração que permite descrever os elementos da infraestrutura subjacente a uma rede virtual no contexto SDN, e que é consumida ao longo da execução da plataforma elaborada. A plataforma estabelecida permite a concepção de um novo modelo de roteamento, o serviço RFAgg, que possibilita a operação unificada de diferentes dispositivos de encaminhamento organizados em uma rede de malha completa. Para se atingir tal objetivo, as propostas e conceitos apresentados neste trabalho foram inseridos no código da arquitetura RouteFlow, grande parte dos quais foram incorporados à versão oficial da ferramenta.

Outras frentes de pesquisa se abrem em decorrência dos resultados conseguidos.

Dentre algumas, a incorporação do comportamento *hot-potato* ao serviço RFAgg e, consequentemente, um estudo da viabilidade de se obter uma arquitetura onde o comportamento *hot-potato* seja imune a instabilidades na rede física e, por conta disto, não afete a política de roteamento do AS. Além disso, almeja-se também a implementação de novos serviços para a plataforma RouteFlow, de maneira a ser possível a aplicação de lógicas inteiramente novas ao encaminhamento de tráfego de dados na Internet.

Referências

Alves, N., de Albuquerque, M. P., de Albuquerque, M. P., and de Assis, J. T. (2010). Topologia e modelagem relacional da internet brasileira. Technical Report CBPF-NT-004/04, Centro Brasileiro de Pesquisas Físicas.

Caesar, M., Caldwell, D., Feamster, N., Rexford, J., Shaikh, A., and van der Merwe, J. (2005). Design and implementation of a routing control platform. In *NSDI '05*.

Caesar, M., Casado, M., Koponen, T., Rexford, J., and Shenker, S. (2010). Dynamic route recomputation considered harmful. *SIGCOMM Comput. Commun. Rev.*, 40:66–71.

Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Huici, F., and Mathy, L. (2008). Towards high performance virtual routers on commodity hardware. In *CoNEXT '08*.

Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. (2008). Nox: towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38:105–110.

Keller, E. and Rexford, J. (2010). The "platform as a service"model for networking. In *INM/WREN'10*.

Nascimento, M. R., Rothenberg, C. E., Salvador, M. R., Corrêa, C. N. A., de Lucena, S. C., and Magalhães, M. F. (2011). Virtual routers as a service: the routeflow approach leveraging software-defined networks. In *CFI '11*.

of Stanford, U. (2010). Beacon: a java-based openflow control platform. `http://www.beaconcontroller.net/`.

Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. *Computer*, 40:38–45.

Park, J. H., Oliveira, R., Amante, S., McPherson, D., and Zhang, L. (2010). Bgp route reflection revisited. CS Technical Report 100006, UCLA.

Sharafat, A. R., Das, S., Parulkar, G., and McKeown, N. (2011). Mpls-te and mpls vpns with openflow. *SIGCOMM Comput. Commun. Rev.*, 41:452–453.

Teixeira, R., Shaikh, A., Griffin, T., and Rexford, J. (2004). Dynamics of hot-potato routing in ip networks. *SIGMETRICS Perform. Eval. Rev.*, 32:307–319.

Udugama, A. (2006). Manipulating the network environment using rtnetlink. *Linux J.*

Verkaik, P., Pei, D., Scholl, T., Shaikh, A., Snoeren, A. C., and van der Merwe, J. E. (2007). Wresting control from bgp: scalable fine-grained route control. In *ATC '07*.

Zhang-Shen, R., Wang, Y., and Rexford, J. (2008). Atomic routing theory: Making an as route like a single node. Technical Report TR-827-08, Princeton University.