ROUTEOPS: Orquestração de Rotas Centrada na Operação de Sistemas Autônomos

Rafael S. Guimarães[1,2], Magnos Martinello[1], Cristina K. Dominicini[1,2]
Dione S. A. Lima[1], Rodolfo S. Villaça[1], Moisés Renato N. Ribeiro[1] *

[1] Núcleo de Estudos em Redes Definidas por Software (NERDS)
Universidade Federal do Espírito Santo (UFES) – Vitória, ES

[2]Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo (IFES)
Campus Cachoeiro/ES, Serra/ES – Brazil

rafaelg@ifes.edu.br, magnos@inf.ufes.br, cristina.dominicini@ifes.edu.br
dione.sousa@gmail.com, rodolfo.villaca@ufes.br, moises@ele.ufes.br

Abstract. *The Internet is formed by the union of different autonomous systems (ASes), which use the BGP protocol for exchanging routing information. In each AS, the operators make use of archaic and obscure mechanisms for defining intra-AS and inter-AS policies. In this context, this paper presents ROUTEOPS: an architecture centered on ASes operation for orchestrating routes. The proposal is based on the use of a SDN controller with centralized view of the AS that assists in defining more expressive policies and facilitate their implementation. The centralized vision is guaranteed by integrating ROUTEOPS with eBGP announcements from neighboring ASes; while control and greater expressiveness of routing policies are achieved with OpenFlow and the centralization of iBGP announcements (intra-AS). The proposed architecture is implemented in Mininet and several experiments were executed as proof of concept, in order to demonstrate its feasibility and benefits.*

Resumo. *A Internet é formada pela união de diversos sistemas autônomos (ASes), que utilizam o protocolo BGP para troca de informações de rotas. Em cada AS, seus operadores fazem uso de mecanismos arcaicos e obscuros na definição de políticas intra-AS e inter-AS. Nesse contexto, este artigo apresenta a ROUTEOPS: uma arquitetura para orquestração de rotas centrada na operação de ASes. A proposta se baseia na utilização de um controlador SDN com visão centralizada do AS que auxilia na definição de políticas mais expressivas e facilita a aplicação das mesmas. A visão centralizada é garantida pela integração da ROUTEOPS com os anúncios eBGP provenientes dos ASes vizinhos; o controle e a maior expressividade das políticas de roteamento são alcançados com o uso do OpenFlow e da centralização dos anúncios iBGP (intra-AS). A arquitetura proposta é implementada no Mininet e vários experimentos foram executados como prova de conceito, visando demonstrar sua viabilidade e as vantagens de sua utilização.*

*Este trabalho tem recebido financiamento do projeto Horizon 2020 da União Européia para pesquisa, desenvolvimento tecnológico e demonstração sob no. 688941 (FUTEBOL), assim como do Ministério Brasileiro da Ciência, Tecnologia e Inovação (MCTI) por meio da RNP e do CTIC. Além disso, gostaríamos de agradecer o financiamento do CNPq sob no. 456143/2014-9 e 449369/20145, e da FAPES sob no. 524/2015.

1. Introdução

A Internet é formada pela união de aproximadamente 12000 Sistemas Autônomos (ASes, *Autonomous Systems*) independentes. Nesta arquitetura, cada AS possui suas próprias políticas, que são aplicadas à rede mediante distribuição de prefixos IP *inter-AS* usando o protocolo BGP (*Border Gateway Protocol*). Embora o BGP seja o principal protocolo existente para se anunciar rotas entre os sistemas autônomos, a não existência de uma padronização na definição de políticas de cooperação entre os AS (*inter-AS*) dificulta a sua operação interna (*intra-AS*).

As políticas de roteamento *inter-AS* e *intra-AS* precisam ser frequentemente modificadas em função de informações espaciais ("de onde vêm os dados?" ou "para onde eles vão?"), temporais ("quando?") e dos acordos existentes entre os AS (bilaterais ou multilaterais) [Akashi et al. 2006]. Por ser pouco flexível e de difícil gerenciamento, o modelo atual de cooperação faz com que os operadores de rede acabem tendo que usar mecanismos arcaicos e obscuros, tais como a aplicação de filtros baseados somente em prefixos IP, para modificar suas políticas de divulgação. Em geral esses filtros possuem pouca expressividade, são de difícil compreensão e precisam ser corretamente configurados em cada roteador envolvido na política que se quer expressar.

Por geralmente operarem um AS, os Provedores de Serviço de Internet (ISP, *Internet Service Provider*) são as principais entidades interessadas no desenvolvimento de soluções que facilitem a operação de redes baseadas no protocolo BGP. Neles a arquitetura de rede geralmente é composta tanto de arranjos *inter-AS*, com a utilização de eBGP (*External Border Gateway Protocol*), quanto de arranjos *intra-AS*, com a utilização do iBGP (*Internal Border Gateway Protocol*). O iBGP é usado para disseminar, no interior de um AS, informações de rotas aprendidas externamente por meio do eBGP. Além dos problemas citados anteriormente referentes à pouca expressividade das regras e à obscuridade na definição das políticas, por ser um protocolo baseado em um algoritmo de vetor de distância a divulgação de rotas somente entre vizinhos iBGP limita a visão geral dos roteadores. Isso afeta, por exemplo, a visualização de todas as possíveis rotas para um determinado destino a partir de um ponto no interior do AS. Além disso, o encaminhamento sempre é baseado no prefixo IP de destino, limitando a sua expressividade e dificultando a criação de regras específicas para determinados tipos de serviço (vídeo sob demanda ou armazenamento na nuvem), por exemplo.

Este artigo defende a ideia de que uma visão centralizada dos anúncios provenientes dos ASes vizinhos, o controle centralizado dos roteadores no interior do AS e uma maior expressividade na criação das regras de encaminhamento, são elementos centrais para avançar na orquestração de rotas na perspectiva de um AS. Nesse contexto, a separação entre as funções de encaminhamento e controle, existente nas Redes Definidas por Software (SDN, *Software Defined Networks*), oferece aos pesquisadores e operadores de rede uma excelente oportunidade para contribuir na definição e implementação das políticas de roteamento [Rothenberg et al. 2012, Gupta et al. 2014] nos ISPs.

Desta forma, este artigo apresenta a ROUTEOPS: uma arquitetura inovadora para orquestração de rotas centrada na operação de sistemas autônomos. Contrariamente à expressividade limitada presente nas arquiteturas tradicionais de implantação do protocolo BGP, a ROUTEOPS avança o estado da arte i) ao oferecer expressividade mais clara e ampla no controle e divulgação interna dos anúncios de rotas recebidos pelo AS; ii) ao

ampliar a granularidade do controle das políticas de roteamento *intra-AS*, não existente nas formas atuais de implantação de roteamento dinâmico. Por meio de um controlador SDN com uma visão logicamente centralizada da infra-estrutura interna do AS, pode-se auxiliar na definição de políticas mais expressivas e facilitar a aplicação dessas políticas no interior do AS. A visão externa (*inter-AS*) é garantida pela integração da ROUTEOPS com os anúncios eBGP provenientes dos ASes vizinhos. O uso de iBGP e eBGP garante a interoperabilidade da arquitetura ROUTEOPS com roteadores legados e com os AS vizinhos que não fazem uso da ROUTEOPS.

A arquitetura ROUTEOPS é implementada em ambiente emulado baseado no Mininet[1] e vários experimentos foram executados como prova de conceito, visando demonstrar a viabilidade de implementação da arquitetura proposta e as vantagens de sua utilização em um AS. O artigo é estruturado da seguinte forma: Na Seção 2, são apresentadas as características dos anúncios *intra-AS* e aborda os trabalhos relacionados à proposta. A Seção 3 detalha as características da arquitetura ROUTEOPS. A Seção 4 apresenta os resultados como prova de conceito. A Seção 5 conclui o artigo e apresenta os trabalhos futuros.

2. Roteamento BGP e Trabalhos Relacionados

Em um ISP que opera um AS, o protocolo BGP é usado para descobrir rotas a serem percorridas por um pacote para um determinado destino.O iBGP é usado como mecanismo de distribuição das rotas, recebidas via anúncios externos, para o interior do AS. Uma diferença crucial entre eBGP e iBGP, basicamente, é o uso de mecanismos diferentes para a prevenção de "*looping*" no anúncio de suas rotas. Neste caso, o eBGP simplesmente olha para o atributo AS_PATH, que contém a lista de ASes e compara com o ASN (*Autonomous System Number*) de seus vizinhos. O iBGP, por sua vez, utiliza outros meios para lidar com este problema, uma vez que todos os seus vizinhos possuem o mesmo ASN. Contudo, as sessões iBGP podem ser estabelecidas de duas maneiras: *full-mesh* ou centralizados em um único roteador.

A utilização de conexões *full-mesh* no iBGP não é escalável, pois o número de sessões BGP entre os roteadores será de $\frac{\alpha(\alpha-1)}{2}$, sendo α o número de roteadores no ISP. Uma alternativa para este tipo de conexão é a utilização de Reflexão de Rotas (*Route Reflection*) [Society 2006], uma abordagem que habilita um determinado roteador (*Route Reflector*) a atuar como uma espécie de concentrador de rotas, repassando o anúncio de rotas recebidas por ele para outros pares de roteadores iBGP vizinhos. Para a utilização de *Route Reflection*, os operadores de rede devem efetuar configurações adicionais no *Route Reflector* visando delimitar o escopo da reflexão, evitando problemas de "*looping*" nos anúncios para os seus vizinhos. Por outro lado, a reflexão de rotas torna a visão mais restrita em relação aos anúncios de todos os ASes, uma vez que serão refletidas somente as melhores rotas na visão de um determinado roteador (*Route Reflector*).

Em resumo, o uso do iBGP impõe fortes restrições na orquestração de rotas na rede interna de um ISP devido a dois fatores principais: i) a inviabilidade de ligações *full-mesh* entre os roteadores de um AS de médio ou grande porte, o que daria uma maior visão da infraestrutura da rede e; ii) as limitações do mecanismo de reflexão de rotas usando iBGP, que impede a orquestração de rotas com critérios customizados de acordo com a

[1] http://mininet.org/

visão do operador da rede. Nesse contexto, a seguir serão apresentados alguns trabalhos importantes relacionados à ROUTEOPS.

O RCP [Feamster et al. 2004] é pioneiro na criação de uma visão logicamente centralizada do sistema de roteamento, personalizando a distribuição interna dos anúncios com a substituição do método de reflexão de rotas do iBGP. Além disso, a abordagem do RCP simplifica a configuração e seleção da melhor rota: ao invés de utilizar-se dos atributos de seleção do BGP, propõe uma nova camada de interação *inter-AS*. A ROUTEOPS se inspira no RCP no tratamento do roteamento *intra-AS*, substituindo o método convencional de reflexão de rotas pela centralização do controle interno dos anúncios. A utilização, pela ROUTEOPS, de um controlador SDN baseado no protocolo OpenFlow traz uma maior granularidade na configuração do encaminhamento de pacotes e habilita a engenharia de tráfego em decorrência da interação direta com o plano de dados. Mais especificamente, a ROUTEOPS se diferencia do RCP principalmente por não criar uma camada de comunicação *inter-AS*, respeitando a autonomia dos ASes na divulgação de prefixos de rede.

O SDX [Gupta et al. 2014] foi utilizado como base para a criação da ROUTEOPS e também propõe a existência de um controle centralizado, porém com visão completa do encaminhamento *inter-AS*. Para isso, o SDX também faz uso do protocolo OpenFlow para contrapor as principais limitações do roteamento BGP, que são: i) encaminhamento baseado apenas no endereço de destino; ii) influencia apenas sobre seus vizinhos e; iii) expressão indireta das políticas de roteamento. Assim como na ROUTEOPS, o uso do OpenFlow garante uma maior expressividade na criação e manipulação das políticas de cada participante da rede. Entretanto, o SDX delimita seu escopo de controle aos IXPs (*Internet Exchange Points*) e comunicação *inter-AS*, enquanto a *ROUTEOPS* se diferencia por ter como foco os sistemas autônomos, tratando o anúncio de rotas *intra-AS* com uso do iBGP como mecanismo de transporte.

Na visão de uma estrutura *intra-AS*, o BTSDN [Pingping Lin 2014] utiliza informações obtidas nos roteadores de borda para influenciar o encaminhamento *intra-AS*, utilizando o OpenFlow como elemento de configuração de regras de encaminhamento, assim como a ROUTEOPS. A finalidade principal desse arranjo é obter uma cooperação entre uma rede OpenFlow e os roteadores de borda legados, sem alterá-los, com a criação de regras de fluxo auxiliadas pela visão interna dos anúncios. Através da utilização do iBGP, o controlador não tem apenas informações de anúncios *intra-AS*, mas também informações *inter-AS* de forma indireta. A principal diferença entre a arquitetura *ROUTEOPS* e o BTSDN, é que a primeira trabalha de forma ativa na divulgação dos anúncios *intra-AS*, permitindo a divulgação de anúncios personalizados para determinados roteadores, enquanto a última apenas faz o mapeamento das rotas em regras de encaminhamento OpenFlow de forma passiva.

O RouteFlow [Nascimento et al. 2011] é uma abordagem que faz uso da separação entre plano de dados e controle por meio do protocolo OpenFlow. O RouteFlow, assim como o BTSDN, também faz um mapeamento de rotas em regras de encaminhamento OpenFlow, utilizando-se de máquinas virtuais e *switches OpenFlow* para implementação do plano de dados. No Routeflow, o *switch* terá as mesmas funções de um roteador convencional, através da realização do encaminhamento usando reescrita do endereço de MAC de destino. Com isso, o RouteFlow habilita a experimentação de diferentes for-

mas de encaminhamento no plano de dados associado aos anúncios BGP. A principal diferenciação para a ROUTEOPS é que no RouteFlow existe uma rede sobreposta de roteadores virtuais mapeados em *switches* OpenFlow que realizarão o encaminhamento dos pacotes e, no caso do ROUTEOPS, o switch Openflow é utilizado apenas para melhorar a granularidade do encaminhamento interno, em conjunto com a visão inter-AS proporcionada pelo BGP.

Após uma breve revisão dos principais trabalhos relacionados é importante reforçar as duas principais contribuições da ROUTEOPS: i) a disponibilização de uma maior expressividade no controle e divulgação interna dos anúncios de rotas recebidos pelos ASes vizinhos; ii) a ampliação da granularidade do controle das políticas de roteamento *intra-AS*, não existente nas formas atuais de implantação de roteamento dinâmico. Como contribuição secundária, mas não menos importante, a proposta da ROUTEOPS também propicia um ambiente adequado para experimentação de soluções de encaminhamento inovadoras no interior dos ASes, com utilização do protocolo OpenFlow e *switches* SDN no núcleo dos ASes e roteadores legados nas bordas.

3. ROUTEOPS

A arquitetura proposta baseia-se na integração, de forma centralizada, entre a arquitetura BGP e o protocolo OpenFlow[McKeown et al. 2008], de forma que a primeira fornece acesso aos anúncios intra e inter-domínios de toda a infra-estrutura de um ISP e o último permite a manipulação do plano de dados. A Figura 1 mostra os módulos da arquitetura responsáveis pela construção e manipulação de políticas de roteamento, que serão descritos a seguir.

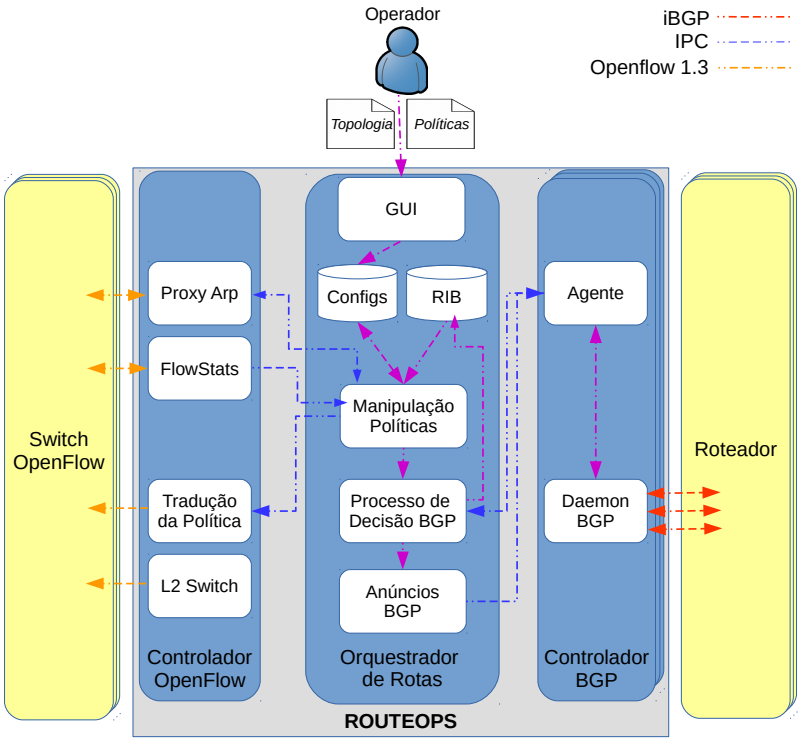
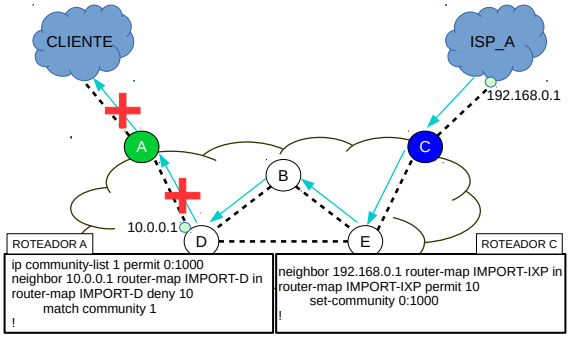


Figura 1. Arquitetura ROUTEOPS

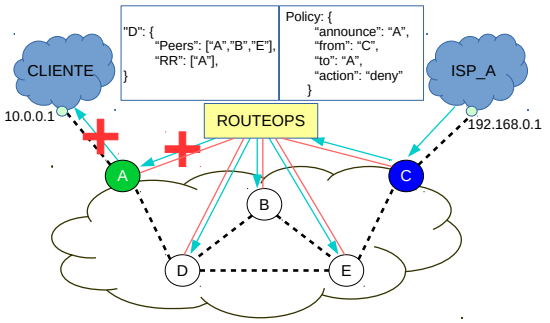
3.1. Orquestrador de Rotas

Suponha que um operador de rede deseja configurar uma política que não aceitará rotas anunciadas pelo *ISP_A*. Na Figura 2(a) o operador de rede deverá configurar em C que as rotas anunciadas por *ISP_A* sejam associadas a uma determinada "*community*". No BGP, "*community*" é um atributo no qual os valores podem estar associados a uma rota, utilizada como uma espécie de marcação para os operadores. No roteador A , baseando-se no valor da "*community*" configurado em C, um filtro deverá ser aplicado. No exemplo, nota-se que uma inconsistência nas configurações dos roteadores invalidará a política de roteamento, ou seja, deverá existir uma sincronia entre os valores de "*community*" definidos no roteador C e A para que este tipo de política funcione. A *configuração topológica* no BGP é consequência das ligações entre os roteadores, que só trocam informações com seus vizinhos.

Na configuração topológica da ROUTEOPS é preciso definir os endereços MAC (*Media Access Control*), IP (*Internet Protocol*), as ligações lógicas entre os roteadores e a política de reflexão das rotas para cada roteador do AS. Para a criação de políticas, o operador poderá criar regras que manipulam o processo de decisão dos anúncios no BGP e regras específicas que diversificam o encaminhamento interno dos pacotes. Por exemplo, na Figura 2(b), para o roteador D foram definidas ligações entre os roteadores (*Peers*) A, B e E e *Router Reflector* (RR) para A, o que significa que A receberá anúncios de B e E mesmo não sendo seus vizinhos.



(a) Arquiteturas Convencionais



(b) ROUTEOPS

Figura 2. Exemplo de expressividade de políticas utilizando método tradicionais em comparação com as políticas definidas no ROUTEOPS

Voltando ao exemplo onde o operador deseja configurar uma política que não aceitará rotas anunciadas pelo *ISP_A*, vamos ilustrar como essa política pode ser expressada de forma simples com a utilização da ROUTEOPS. Na Figura 2(b), descrevemos que os anúncios originados em C e destinados a A serão negados ("action deny").

Como podemos observar, a definição de uma determinada política no modelo tradicional é obscura, conforme observado na Figura 2(a), e a determinação da política é descentralizada com uma visão limitada dos anúncios de todas as rotas. Com a utilização da ROUTEOPS, expressamos a topologia e as políticas diretamente no módulo *Orquestrador de Rotas* através de uma *GUI (Graphical User Interface)* de acesso às configurações e políticas. Desta forma o operador terá um leque de opções de políticas que diversificam o encaminhamento dos anúncios para um determinado roteador, entregando a melhor rota para aquele roteador. Tudo isso em função de uma visão e controle centralizados e a possibilidade de orquestração destes anúncios, com base em critérios definidos pelo operador.

Além disso, o módulo *Orquestrador de Rotas* recebe e envia mensagens para o módulo *Controlador BGP* e para o módulo *Controlador OpenFlow*. Na comunicação com o módulo *Controlador BGP*, o componente *Processo de Decisão BGP* é responsável por receber todos os anúncios e, em seguida, tomar uma decisão sobre qual deles oferece a "melhor" rota para cada destino. Essas melhores rotas são colocados em uma base de dados, denominada de RIB (*Routing Information Base*). No entanto, a definição da "melhor" rota envolve diversos critérios além de observar o caminho mais curto.

O componente *Processo de Decisão BGP*, após definir as melhores rotas, deverá encaminhar os anúncios para o componente *Anúncios BGP*, utilizando, de forma similar, dos critérios de reflexão de rotas da *RFC4456* utilizados no BGP. O componente *Anúncios BGP*, por sua vez, irá encaminhar esses anúncios para o módulo *Controlador BGP* que, por fim, envia os anúncios para os roteadores. A forma como acontecerá a reflexão de rotas deve ser definida e configurada de acordo com as políticas específicas daquele AS. O componente *Manipulação de Políticas*, a partir das políticas pré-definidas pelo operador de rede, influencia no processo de decisão do protocolo BGP. Essa influência acontece por meio da interação com o componente *Processo de Decisão BGP* e *Tradução da Política* do módulo *Controlador OpenFlow*. A interação com o operador de rede é feita por meio de uma interface gráfica, componente *GUI*, que permite a inserção de configurações de topologia e políticas na base de dados do módulo *Orquestrador de Rotas*.

3.2. Controlador BGP

No módulo *Controlador BGP*, o componente *Daemon BGP* é responsável pela conectividade iBGP (Internal BGP) com os roteadores legados. Portanto, este componente será responsável por estabelecer uma sessão entre os pares BGP (*Peers*) e trocar informações sobre atualização de rotas e notificações de erros.

O componente *Agente* é responsável por receber mensagens do componente *Daemon BGP* e enviar mensagens para o módulo *Orquestrador de Rotas* utilizando o formato JSON[2]. Além disso, o componente *Agente* recebe as mensagens do módulo *Orquestrador de Rotas* e as encaminha para o componente *Daemon BGP* em formato adequado.

[2]http://www.json.org

3.3. Controlador OpenFlow

No módulo *Controlador OpenFlow*, responsável por enviar e receber mensagens OpenFlow para um determinado *switch*, o componente *Tradução da Política* é responsável por traduzir as políticas definidas pelo módulo *Orquestrador de Rotas* em regras OpenFlow na versão 1.3. Além disso, o componente *FlowStats*, recebe dos *switches* informações relacionadas aos fluxos que serão entregues ao componente *Manipulação de Políticas* do módulo *Orquestrador de Rotas*. Já o componente *Proxy ARP* responde as requisições *ARP* obedecendo às determinações expressas nas políticas definidas no módulo *Orquestrador de Rotas*.

3.4. Implementação

A implementação do protótipo descrito neste artigo foi dividida em três partes: Controlador OpenFlow, Controlador BGP e Plano de Gerência de Rotas e Políticas (*Router Server*). Cada item foi referenciado como um módulo na arquitetura proposta. A comunicação entre os módulos será realizada através de mensagens IPC (*Inter-Process Communication*), utilizando o banco de dados distribuído NoSQL MongoDB[3] versão 2.6.5. Além disso, essa base de dados NoSQL é usada para implementar uma fila de mensagens JSON entre os módulos, permitindo a comunicação de perda de desempenho e facilitando o gerenciamento de falhas, depuração e monitoramento.

No módulo *Controlador OpenFlow* foi utilizado o controlador Ryu versão 3.23. Nele foram implementados os componentes *Tradução das Políticas*, *L2 Switch*, *FlowStats* e *Proxy ARP*. As principais tarefas associadas a este módulo são: interpretação das políticas, criação de regras de fluxo no plano de dados, respostas ARP_REPLY e coleta de informações de fluxos.

Na implementação do módulo *Orquestrador de Rotas*, foi utilizado a linguagem Python em conjunto com a biblioteca PyMongo[4] para integração com o MongoDB, responsável pela leitura das mensagens recebidas e enviadas para os outros módulos e pela persistência das informações de rotas, políticas e expressões topológicas. Fazem parte deste módulo os componentes *Processo de Decisão BGP*, *Anúncios BGP* e *Manipulação das Políticas*. As principais funcionalidades deste módulo são: armazenamento dos anúncios de rotas e anúncio customizado de rotas com base em uma determinada política.

No módulo *Controlador BGP*, foi utilizado, como Daemon BGP, o *ExaBGP*[5] e, como *Agente*, um software escrito em Python que se comunica diretamente com o *ExaBGP* para habilitar a troca de mensagens com o módulo *Orquestrador de Rotas*.

4. Prova de Conceito

Este seção apresenta o ambiente de experimentação que foi utilizado para a avaliação do protótipo e os resultados obtidos nos experimentos como prova de conceito. Para uma melhor organização da apresentação da prova de conceito, realizou-se a divisão em 3(três) subseções. A primeira (seção 4.1), descreve as características do ambiente de experimentação. Em seguida (seções 4.2 e 4.3), avaliamos o funcionamento do protótipo

[3]https://www.mongodb.org

[4]https://api.mongodb.org/python/

[5]https://github.com/Exa-Networks/exabgp

em determinadas topologias e políticas. Neste caso, apresentamos resultados medindo a vazão de fluxos na visão de operação do AS. Os casos de uso servem para ilustrar a orquestração de rotas da ROUTEOPS.

4.1. Ambiente de Experimentação

Para validar a proposta, utilizamos como ambiente de emulação o Mininet[6] na versão 2.1.0p2 com uma extensão que provê serviços de forma simplificada, denominada de MiniNext[7] na versão 1.10. Esta versão do Mininet provê suporte ao Openflow 1.3, utilizada como base no protótipo. No plano de dados, foi utilizado como máquina de encaminhamento (*Software Switch*), o Open vSwitch[8] na versão 2.3.2. Além disso, todos os enlaces foram configurados com limite de 1Gbps.

Portanto, em uma dada topologia de ISP operando seu próprio AS, cada roteador executa o software *Quagga*[9] na versão 0.99.22.4 para o serviço (daemon) que implementa o protocolo BGP. Para a geração e análise do tráfego durante os experimentos foi utilizada a ferramenta Iperf[10] na versão 3.0.4 com os protocolos TCP e UDP.

4.2. Orquestração de rotas de entrada e saída

O objetivo do primeiro experimento é mostrar uma regra de reflexão de rotas customizadas que impacta na vazão de 2(dois) fluxos de entrada. A topologia utilizada é ilustrada na figura 3, com fluxos TCP com origem no $AS150$ e com destinos para o $AS300$ e para o roteador $R03$ contido no $AS65000$. Utilizamos 3(três) roteadores trocando mensagens iBGP e ligados a um *Switch Openflow* e ao servidor ROUTEOPS, fazendo o papel de um ISP, e 4(quatro) roteadores que trocam mensagens eBGP para realizar troca de anúncio de rotas entre ASes distintos. O $AS300$, exclusivamente, é um cliente, do ISP de exemplo, ligado ao roteador $R02$.

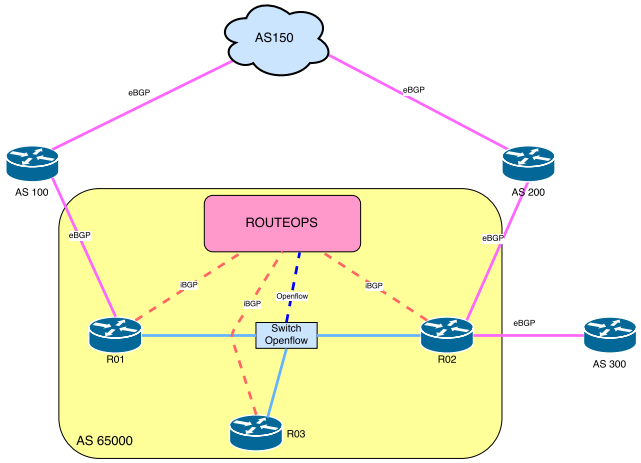


Figura 3. Topologia Cenário 01 e 02

[6]http://mininet.org/
[7]http://mininext.uscnsl.net
[8]http://openvswitch.org/
[9]http://www.nongnu.org/quagga/
[10]https://iperf.fr/

Originalmente, o roteador $R03$ está configurado para refletir suas rotas tanto para $R01$ quanto para $R02$. Estas informações chegam até o $AS150$ via anúncios eBGP que, por sua vez, seleciona o melhor caminho por meio do processo de decisão BGP. Desta forma, os fluxos originados do $AS150$ com destino ao $AS300$(Fluxo 1 na Figura 4) seguia o caminho $AS150$-$AS200$-$R02$-$AS300$. Já os fluxos originados do $AS150$ com destino ao $R03$(Fluxo 2 na Figura 4) seguia o caminho $AS150$-$AS200$-$R02$-$R03$. Assim, como é possível ver na Figura 4 no período de 0s até 200s, a vazão dos fluxos é limitada pela sobrecarga do enlace entre $AS200$ e $R02$. Para resolver tal problema, no instante 200s, o operador insere uma política na *GUI* de acesso ao protótipo para realizar uma reflexão de rotas customizada em que $R03$ reflete suas rotas apenas para $R01$. Desta maneira, conforme ilustrado na Figura 4, o Fluxo 2 com destino para $R03$ passa a ser encaminhado por meio do caminho $AS150$-$AS100$-$R01$-$R03$, liberando o enlace que estava sobrecarregado e aumentando a vazão de ambos os fluxos. Nota-se ainda que houve um tempo de convergência para que cada sistema autônomo passe a interpretar o anúncio das novos anúncios de rotas e a mudança seja efetivada.

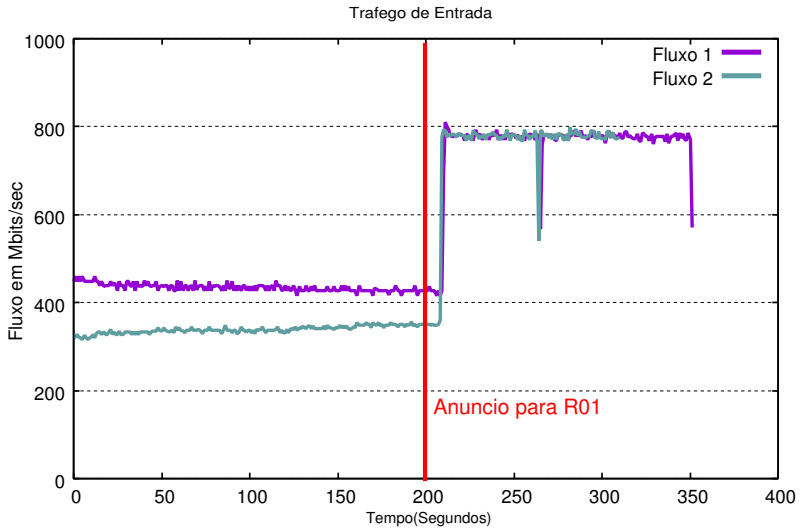


Figura 4. Cenário 01 - Tráfego de entrada

Neste segundo experimento, considera-se como base os fluxos de saída: Fluxo 1 com origem no $AS300$, Fluxo 2 com origem no $R03$ e ambos com destino para o $AS150$. Inicialmente, tanto o roteador $R01$ e $R02$ estão configurados para refletir suas rotas para o $R03$. Desta forma, quando $R03$ deseja encaminhar fluxo para o destino $AS150$ ele utiliza o $R02$ em função do processo de decisão BGP. De forma similar, o $AS300$ quanto deseja encaminhar fluxos para o $AS150$ também escolherá a rota que passe por $R02$. Como consequência, a Figura 5 mostra que durante o período 0s ao 57s a vazão esta limitada devido a sobrecarga do enlace entre $AS200$ e $R02$.

Para resolver tal problema, no instante 57s, o operador insere uma política na *GUI* de acesso ao protótipo para realizar uma reflexão de rotas customizada onde apenas rotas oriundas de $R01$ sejam refletidas para $R03$. Desta maneira, conforme ilustrado na Figura 5, o Fluxo 2 com origem para $R03$ passa a ser encaminhado por meio do caminho $R03$-$R01$-$AS100$-$AS150$, liberando o enlace que estava sobrecarregado e aumentando a vazão de ambos os fluxos. Nota-se ainda que houve um tempo de convergência menor do que o

cenário anterior, já que a mudança do anúncio precisa ser aplicada apenas em $R03$.

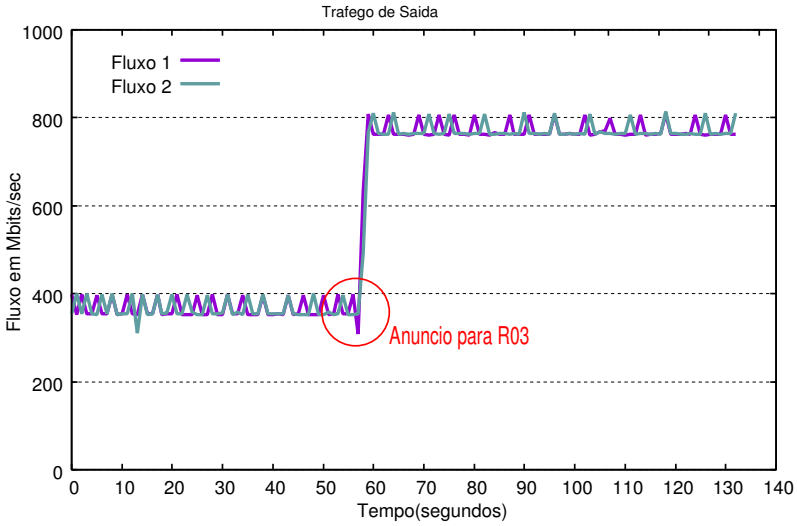


Figura 5. Cenário 02 - Tráfego de saída

4.3. Orquestração de anúncios a partir da monitoração de fluxos

Serviços de vídeo com necessidade de uma alta largura de banda como o YouTube e Netflix, oferecem uma grande demanda de tráfego em relação ao volume global de tráfego, de modo que os ISP estão cada vez mais interessados em trocar tráfego com as aplicações específicas (CDNs) observando sempre o Custo X Benefício para se conseguir realizar estes acordos. De fato, o BGP não torna esta política simples de ser implementada. Um ISP pode determinar o encaminhamento de um determinado fluxo em detrimento da classificação dos serviços atrelados. Para isso, ele deverá identificar o tráfego relevante e efetuar e redirecionar o tráfego para um caminho especial. Por exemplo, um ISP necessitará configurar em seus roteadores de borda diversas tabelas de roteamento que serão utilizadas para uma determinada aplicação e outra tabela de roteamento para as demais aplicações. Este tipo de abordagem cria tabelas com rotas adicionais que sobrecarregam ainda mais a tabela de roteamento e ainda acrescentam uma obscuridade na configuração e classificação dos serviços[Gupta et al. 2014].

Neste experimento é proposta uma solução para esse problema e analisados os impactos de vazão quando um anúncio é feito de forma diversificada por fluxos de cada serviço. A topologia do cenário deste experimento é ilustrada pela Figura 6. Utilizamos 4(quatro) roteadores trocando mensagens iBGP e ligados a um *Switch Openflow* e ao servidor ROUTEOPS, fazendo o papel de um ISP, e 4(quatro) roteadores que trocam mensagens eBGP para realizar troca de anúncio de rotas entre ASes distintos. O $AS300$, exclusivamente, é um cliente ligado aos roteadores $R03$ e $R04$.

Neste cenário, foram configurados dois fluxos com serviços distintos, ambos com origem no $AS150$ e destino para o $AS300$, que competem pelos recursos do meio: o Fluxo 1 do Serviço 1 utiliza o protocolo UDP de forma contínua tentando consumir toda a banda disponível e o Fluxo 2 do Serviço 2 utiliza o protocolo TCP com rajadas de transmissão. Inicialmente, o roteador $AS300$ tem sessões eBGP estabelecidas com os roteadores $R03$ e $R04$, não existe políticas específicas para os anúncios e os dois fluxos são encaminhados

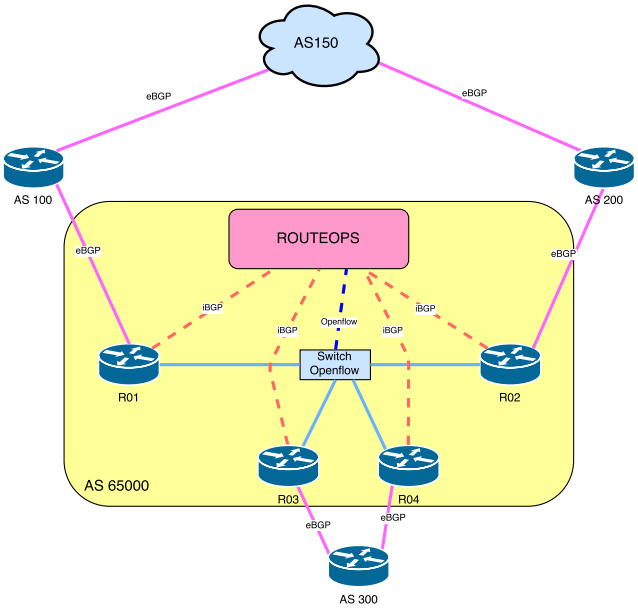


Figura 6. Topologia Cenário 03

pelo mesmo caminho($AS150$-$AS100$-$R01$-$R03$-$AS300$) porém para prefixos distintos do $AS300$.

Com base em um SLA (*Service Level Agreement*) para o cliente do $AS300$, em que será priorizado o QoS (*Quality of Service*) para o Serviço 1, a política determina como ação um anúncio diversificado na infra-estrutura do ISP caso o serviço em questão não consiga garantir uma determinada vazão mínima para os prefixos anunciados pelo $AS300$. No instante anterior a 100s, a vazão do Fluxo 1 diminui em função da competição pelos recursos de rede com o Fluxo 2, conforme ilustrado na Figura 7. Então, o operador de rede detecta que o SLA do Serviço 1 não está sendo cumprido e, após o instante 100s, informa uma política em que determina um valor mínimo de vazão para o Fluxo 1, que neste caso foi de 512Mbps, e uma ação de diversificação de anúncio para o roteador $R02$, onde o roteador $R04$ será responsável por encaminhar os fluxos relacionados aos prefixos do Serviço 1 para o $AS300$. Não alterou-se nenhuma política em relação ao Serviço 2 de forma que o Fluxo 2 continua a seguir pelo caminho $AS150 - AS100 - R01 - R03 - AS300$. Já o Fluxo 1 do Serviço 1, passa a seguir o caminho $AS150 - AS100 - R02 - R04 - AS300$. A partir disso, observamos que além de obtermos uma melhor vazão para o Fluxo 1, melhoramos também a vazão do Fluxo 2. Desta maneira, mostra-se que observar a vazão de um serviço associado a um determinado prefixo, e realizar uma anúncio diversificado em função dessa observação, garante uma engenharia de tráfego mais expressiva dentro da infraestrutura de um ISP.

Para que essa política seja aplicada, é adicionada uma regra específica de encaminhamento no *Switch Openflow* que conta o número de bytes dos fluxos que correspondem ao padrão: Endereços IPs de Origem $AS150$, Endereços IPs de destinos para $AS300$, porta de origem e protocolo de transporte. O intervalo de amostragem será de 10s. Com essa informação, o controlador terá uma amostra da vazão do Serviço 1 entre os ASes $AS150$ e $AS300$. O tempo de coleta de informações de cada fluxo fica condicionado a não deteriorar o desempenho do plano de dados e do plano de controle [Curtis et al. 2011].

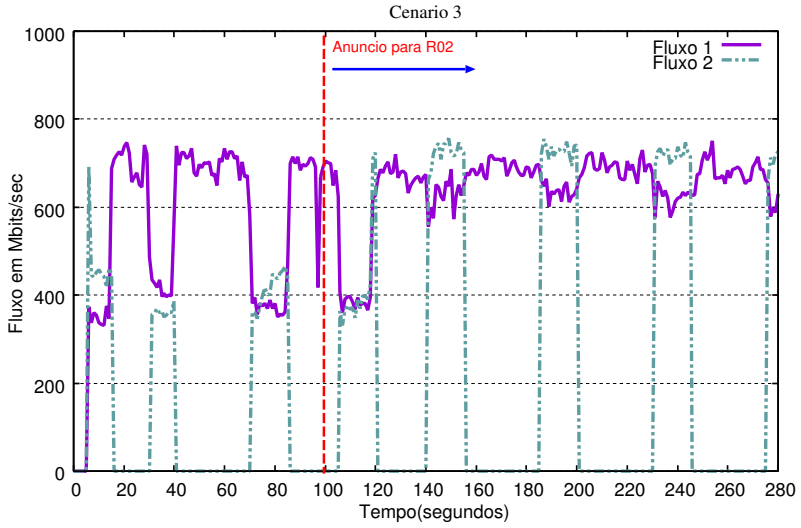


Figura 7. Cenário 03 - Anúncio diversificado com base na análise dos fluxos

5. Conclusões e Trabalhos Futuros

As políticas de roteamento em sistemas autônomos (ASes) são ditadas essencialmente pelo protocolo BGP. O BGP faz uso de mecanismos, tais como a aplicação de filtros baseados somente em prefixos IP, que possuem i) pouca expressividade (arcaicos), ii) são de difícil compreensão (obscuros) e iii) precisam ser corretamente configurados em cada roteador envolvido para que a política seja aplicada na infraestrutura de rede do sistema autônomo (tempo longo de convergência).

Neste contexto, este trabalhado apresenta a ROUTEOPS que é uma arquitetura para orquestração de rotas centrada na operação de sistemas autônomos. ROUTEOPS apoia-se no conceito de SDN com base na visão centralizada provida por um controlador que facilita definir políticas de modo mais claro, garantindo aplicação dessas políticas de modo ágil na infraestrutura interna do AS. A orquestração de rotas é centrada na operação do AS pela integração da ROUTEOPS com os anúncios eBGP provenientes dos ASes vizinhos. Tanto o controle da infra-estrutura interna do AS, quanto a maior expressividade na aplicação das políticas de roteamento são alcançados com o uso de OpenFlow. O protocolo OpenFlow define como será o encaminhamento interno das mensagens iBGP para os roteadores no interior do AS. O uso de iBGP e eBGP garante a interoperabilidade da arquitetura ROUTEOPS com roteadores legados e com os AS vizinhos que não fazem uso da ROUTEOPS.

Como prova de princípio, ROUTEOPS foi implementada em um ambiente de emulação de redes (Mininet). Experimentos demonstraram que ROUTEOPS permite a operação de rotas no AS em tráfegos de entrada e saída no AS que levou a um uso mais equilibrado da infraestrutura do AS. Houve também redução no tempo de convergência para aplicação da política de roteamento intra-AS.

Como trabalhos futuros, pretende-se utilizar a abordagem do trabalho [Alan Chang et al. 2015], no qual realiza uma tratamento misto do encaminhamento através de roteadores convencionais e um *Switch Openflow*, que mostra um melhor desempenho no tempo de convergência do encaminhamento interno. Como outro trabalho

futuro, pretende-se utilizar uma forma de redução na tabela de roteamento proposto pelo trabalho [Luiz Fernando T. de Farias 2014], que permite uma redução das tabelas de encaminhamento (FIB) nos roteadores internos.

Referências

Akashi, O., Fukuda, K., Hirotsu, T., and Sugawara, T. (2006). Policy-based bgp control architecture for autonomous routing management. In *Proceedings of the 2006 SIGCOMM Workshop on Internet Network Management*, INM '06, pages 77–82, New York, NY, USA. ACM.

Alan Chang, M., Holterbach, T., Happe, M., and Vanbever, L. (2015). Supercharge me: Boost router convergence with sdn. *SIGCOMM Comput. Commun. Rev.*, 45(5):341–342.

Curtis, A. R., Mogul, J. C., Tourrilhes, J., Yalagandula, P., Sharma, P., and Banerjee, S. (2011). Devoflow: Scaling flow management for high-performance networks. *SIGCOMM Comput. Commun. Rev.*, 41(4):254–265.

Feamster, N., Balakrishnan, H., Rexford, J., Shaikh, A., and van der Merwe, K. (2004). The Case for Separating Routing from Routers. In *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, Portland, OR.

Gupta, A., Vanbever, L., Shahbaz, M., Donovan, S. P., Schlinker, B., Feamster, N., Rexford, J., Shenker, S., Clark, R., and Katz-Bassett, E. (2014). Sdx: A software defined internet exchange. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 551–562, New York, NY, USA. ACM.

Luiz Fernando T. de Farias, Morganna C. Diniz, S. C. d. L. (2014). Uma abordagem para redução da tabela de encaminhamento sob a ótica da interface de saída dos pacotes. In *XIII Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance 2014)*, Porto Alegre/RS. SBC.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.

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 *Proceedings of the 6th International Conference on Future Internet Technologies*, CFI '11, pages 34–37, New York, NY, USA. ACM.

Pingping Lin, Jun Bi, H. H. (2014). Btsdn: Bgp-based transition for the existing networks to sdn. IEEE 2014, Shanghai. IEEE.

Rothenberg, C. E., Nascimento, M. R., Salvador, M. R., Corrêa, C. N. A., Cunha de Lucena, S., and Raszuk, R. (2012). Revisiting routing control platforms with the eyes and muscles of software-defined networking. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, pages 13–18, New York, NY, USA. ACM.

Society, T. I. (2006). Bgp route reflection: An alternative to full mesh internal bgp (ibgp). Website. `https://www.ietf.org/rfc/rfc4456.txt`.