Universidade Federal do Amazonas
Instituto de Computação
Programa de Pós-graduação em Informática

Ricardo Bennesby da Silva

Inovação no Roteamento Inter-Domínio com Redes Definidas por Software

Manaus-AM
2013

UFAM

Universidade Federal do Amazonas
Instituto de Computação
Programa de Pós-graduação em Informática

Ricardo Bennesby da Silva

Inovação no Roteamento Inter-Domínio com Redes Definidas por Software

Dissertação apresentada ao Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do título de Mestre em Informática.

Orientador: Edjard de Souza Mota

Manaus-AM
2013

Ricardo Bennesby da Silva.

Inovação no Roteamento Inter-Domínio com Redes Definidas por Software

54 páginas

Dissertação (Mestrado) - Instituto de Computação da Universidade Federal do Amazonas.

1. Software-Defined Networking
2. Interdomain Routing
3. BGP

Universidade Federal do Amazonas. Instituto de Computação. Programa de Pós-Graduação em Informática.

Banca Avaliadora:

Prof. Dr.
Dorgival Olavo Guedes Neto

Prof. Dr.
Leandro Silva Galvão de Carvalho

Prof. Dr.
Alexandre Passito de Queiroz

Prof. Dr.
Edjard de Souza Mota
Orientador

À Madalena Cardoso, minha mãe
À Késsia Cris, meu amor

Às vezes, a vida bate com um tijolo na sua cabeça.
Não perca a fé. Estou convencido de que a única
coisa que me permitiu seguir adiante
foi o meu amor pelo que fazia. Você tem que descobrir
o que você ama. Isso é verdadeiro tanto para o seu
trabalho quanto para com as pessoas
que você ama.

Seu trabalho vai preencher uma parte grande da sua vida,
e a única maneira de ficar realmente satisfeito é fazer
o que você acredita ser um ótimo trabalho.
E a única maneira de fazer um excelente trabalho
é amar o que você faz.

Se você ainda não encontrou o que é, continue procurando.
Não sossegue. Assim como todos os assuntos do coração,
você saberá quando encontrar. E, como em qualquer grande
relacionamento, só fica melhor e melhor à medida que os anos
passam. Então continue procurando até você achar.
Não sossegue.

Steve Paul Jobs

Agradecimentos

Agradeço primeiramente a Deus pelo dom da vida e por todas as oportunidades que me foram dadas para crescer pessoal e profissionalmente.

À minha família, Clarita, Madalena, Mara, Marilene, Gabi, Bruna, Lucas, Juliana, Jozimar e Sanches, por todo o apoio e orientações dados em cada decisão tomada em minha vida. Em especial à minha mãe, Madalena Cardoso, pelo amor e doação dedicados a mim. Devo muito a ela pela educação e amor recebidos. Sempre disposta a me ajudar a superar os momentos difíceis. É minha fortaleza; sem ela eu não teria chegado onde estou.

À Késsia Cris, namorada e amiga que levarei para a vida toda. Acompanha meus passos desde antes de iniciar a graduação. Sempre compreensiva e paciente nos momentos difíceis, me deu muita força nessa caminhada. Apoiou-me, aconselhou-me, incentivou-me. Agradeço também à sua família, por todo apoio, em especial a d. Nazaré, sr. Serafim, Sanara e sr. Jorge.

Ao Edjard Mota, meu orientador, por ter me acolhido no grupo de pesquisa a que faço parte, por ter me orientado em projetos de pesquisa ainda na graduação e agora no mestrado. Posso dizer que é um grande orientador, que me ouviu quando tive dificuldades, deu ideias quando preciso, mostrou os caminhos a seguir e esteve sempre presente para ajudar no trabalho. Mostrou a importância de trabalhar em grupo, ressaltando que é em equipe que somos mais fortes e temos conquistas mais significativas. Hoje é mais do que orientador, é um amigo. Agradeço também meus companheiros do nosso grupo de pesquisa, o LabCIA: Alexandre Passito, Paulo César, Yan Brandão e Rodrigo Braga. Passamos por muitas situações e desafios, e a experiência adquirida com eles foi incrível.

À pesquisadora Dra. Monica Nogueira, da Universidade da Carolina do Norte/EUA, pela ajuda em diversos artigos científicos e parte do texto desta dissertação. Além das revisões e correções, sempre ajudava com conselhos valiosos.

Aos meus amigos de faculdade, que estiveram comigo todos estes anos nesse processo de formação. Agradeço em especial ao Flávio Montenegro, Paulo César, Hugo Cunha, Bruno Mandell, Emory Raphael, Bruno Dias e Rodrigo Borges.

A todos os professores do ICOMP/UFAM, pois tiveram papel importante na minha formação em Ciência da Computação.

Ao CNPq, pela bolsa de estudo que me auxiliou financeiramente durante o desenvolvimento deste trabalho.

Aos professores presentes em minha qualificação, prof. Dr. Edjair Mota, prof. Dr. Leandro Galvão e prof. Dr. César Melo, pelas contribuições valiosas para este documento e para a defesa da dissertação.

Aos professores que aceitaram o convite para formarem a banca da minha defesa de dissertação, prof. Dr. Dorgival Guedes (UFMG), prof. Dr. Alexandre Passito e prof. Dr. Leandro Galvão.

Agradeço também aos meus amigos da Paróquia São Pedro, em Petrópolis. Estão comigo desde a minha infância, e inegavelmente são parte importante do que hoje sou.

Resumo

A Internet está organizada em grupos de redes que são gerenciados por domínios ad- ministrativos conhecidos como Sistemas Autônomos (ASes– *Autonomous Systems*). Cada AS emprega suas próprias políticas de roteamento e tem autonomia em relação a outros ASes. A comunicação e coordenação entre estes ASes acontece por meio do protocolo de roteamento interdomínio. O Protocolo de Roteamento de Borda (BGP– *Border Gateway Protocol*) é o protocolo de roteamento interdomínio atualmente utilizado na Internet.

Entretanto, a arquitetura do roteamento interdomínio da Internet tem sofrido poucas mudanças desde sua criação. Ela apresenta problemas complexos de serem resolvidos, devido à dificuldade para a implantação de novas soluções, à dificuldade de entender seu comportamento e dinâmica, e à complexidade de identificar e corrigir falhas. A implantação de novas funcionalidades na arquitetura da Internet é uma tarefa difícil, devido à necessidade de manipular diretamente todos os roteadores. Além disso, é necessário aceitação global de novos protocolos e modificações nos existentes.

Nós observamos que a abordagem Redes Definidas por Software (SDN– *Software-Defined Networking*) poderia ser usada para prover inovação no roteamento interdomínio, mas que faltavam mecanismos para suportar esse tipo de roteamento. Neste trabalho, nós apresentamos um mecanismo capaz de realizar o roteamento interdomínio entre domínios que utilizam o paradigma SDN, chamado *Inter-SDN*. Nós ressaltamos os principais problemas no modelo atual de roteamento interdomínio, descrevemos o componente Inter-SDN, seu comportamento, e apresentamos uma análise experimental. Além disso, nós mostramos como o projeto e a construção de um mecanismo de roteamento interdomínio em SDN são tarefas relativamente simples, e explicamos como nossa solução utiliza as vantagens do SDN para resolver problemas do roteamento interdomínio.

Palavras-chave: Redes Definidas por Software, Roteamento Interdomínio, BGP

Abstract

The Internet is organized on network groups managed by administrative domains known as Autonomous Systems (ASes). Each AS employs its own routing policies and has autonomy in comparison to other ASes. The comunication and coordination between these ASes is made possible by the interdomain routing protocol. The Border Gateway Protocol (BGP)is the interdomain routing protocol currently used on Internet.

However, the Internet's interdomain routing architecture has undergone only minor changes since its inception. It presents issues difficult to solve, because of the barrier to deploy new features, the hardness to understand its behavior and dynamics, and complexity to identify and correct faults. The deployment on Internet architecture is a tough task, because the need to deploy it directly on routers. Besides that, there is a requirement of global acceptance of new protocols and modifications.

We observed that Software Defined Networking (SDN) could be used to provide innovation on interdomain routing, but it lacked mechanisms support this approach. SDN is an emerging paradigm composed by a data plane, a control plane, and an open protocol. On this work we present a mechanism able to perform interdomain routing with domains that deploys SDN paradigm, called *Inter-SDN* Routing Component. We point out the main issues on current interdomain routing, describes the Inter-SDN component, its behavior, and its experimental evaluation. Besides that, we show how prototyping and building of an interdomain mechanism on SDN are tasks relatively simple, and explain how our solution takes advantage of the SDN features to address the issues of the interdomain routing.

Keywords: Software-Defined Networking, Interdomain Routing, BGP

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AS Autonomous System

ASN Autonomous System Number

BGP Border Gateway Protocol

DNS Domain Name System

DDoS Distributed Denial of Service

eBGP External Border Gateway Protocol

EGP Exterior Gateway Protocol

iBGP Internal Border Gateway Protocol

ICMP Internet Control Message Protocol

IGP Interior Gateway Protocol

IP Internet Protocol

IRR Internet Routing Registry

ISP Internet Service Provider

MED Multi Exit Discriminator

MRAI Minimum Route Announcement Interval

NOS Network Operating System

OF OpenFlow

OSPF Open Shortest Path First

RIB Routing Information Base

RPSL Routing Policy Specification Language

SDN Software Defined Networking

TCP Transport Control Protocol

Sumário

Capítulo 1

Introdução

1.1 Motivação

A Internet está atualmente particionada em grupos de redes chamados Sistemas Autônomos (ASes– *Autonomous Systems*). Cada AS é um domínio gerenciado por uma entidade administrativa, com políticas próprias e com autonomia em relação aos demais ASes. O protocolo de roteamento interdomínio utilizado na Internet é conhecido como Protocolo de Roteamento de Borda (BGP– *Border Gateway Protocol*) (Rekhter et al., 2006).

O modelo completamente distribuído de cada AS que da Internet é uma solução que provê escalabilidade. Entretanto, a arquitetura de roteamento interdomínio atual apresenta muitos problemas. As decisões de roteamento são tomadas por roteadores distribuídos em cada AS, cada um com uma visão parcial do domínio. A falta de um controle e conhecimento completo do AS, de sua topologia e dos estados das tabelas de roteamento pode gerar inconsistências como *loops* e perda de informações de roteamento (Vissicchio et al., 2012).

O protocolo BGP (Rekhter et al., 2006) está em funcionamento há três décadas. Como aconteceu com outros protocolos, quando o BGP foi projetado, não havia a percepção de que a Internet cresceria tanto e que a demanda por conexão à rede aumentaria muito rapidamente. Para suprir essas necessidades, novas funcionalidades foram adicionadas ao BGP, fazendo com que se tornasse um protocolo complexo, de comportamento de difícil previsão, o que dificulta a prevenção, identificação e correção de erros. Além disso, qualquer mudança no protocolo requer um longo processo de revisão, padronização e aceitação global para que seja efetivamente implantado ou atualizado. O fato de a Internet apresentar sua arquitetura muito dependente de sua infraestrutura, ou seja, depender da particularidade de seus equipamentos, cria uma dificuldade à inovação (Raghavan et al., 2012).

Uma arquitetura que promove a inovação deve ser extensível e abstrata (Koponen et al., 2011). Uma arquitetura é extensível quando permite que novas funcionalidades ou aplicações sejam adicionadas ou substituídas facilmente, evoluindo de acordo com a necessidade. Uma arquitetura é abstrata quando detalhes de baixo nível são desconsiderados, tornando a criação de aplicações mais rápida e confiável. Qualquer mudança significativa na arquitetura da rede atual envolve custos consideráveis para os fabricantes e para os operadores de rede.

Muitos problemas do protocolo BGP são provenientes da troca de informações de roteamento interdomínio entre os roteadores dentro de um Sistema Autônomo, função desempenhada pela comunicação entre os roteadores BGP dentro do mesmo domínio, conhecida como iBGP (*internal BGP*– BGP interno) (Halabi and Mcpherson, 2000). Apesar de proporcionar escalabilidade, o iBGP introduz anomalias de roteamento e repasse, causadas pela oscilação (instabilidade) do protocolo. Além disso, a habilidade de distribuição correta de informações de roteamento que o iBGP realiza pode ser afetada pela inclusão de uma única sessão iBGP. Técnicas para evitar tais anomalias são muito pesquisadas, como por exemplo, a introdução de refletores de rota (Vissicchio et al., 2012).

Com uma arquitetura logicamente centralizada, com as decisões de roteamento separadas dos roteadores, muitos desses problemas poderiam ser resolvidos. Uma Plataforma de Controle de Rotas (RCP) localizada em cada Sistema Autônomo, logicamente centralizada, com uma visão completa de todo o domínio, com os roteadores apenas com a função de repasse, pode ser mais facilmente configurável, gerenciável e menos suscetível a erros (Feamster et al., 2004).

Uma abordagem atual que provê uma separação maior entre a infraestrutura e a arquitetura da rede e cria novas possibilidades é Redes Definidas por Software (SDN) (McKeown et al., 2008). SDN efetivamente separa a rede em plano de dados e plano de controle e é utilizada em muitos datacenters. Apesar de SDN e outras abordagens recentes terem potencial para promover essa separação entre infraestrutura e arquitetura e possibilitar a inovação arquitetural, de acordo com nossas pesquisas poucas soluções foram propostas para resolver os problemas atuais de roteamento interdomínio com tal abordagem.

O crescente sucesso e adoção da abordagem SDN e suas vantagens (Rexford and Dovrolis, 2010) nos motivaram a criar uma nova abordagem de roteamento interdomínio, utilizando os benefícios do paradigma SDN para facilitar a resolução dos problemas da abordagem atual com baixo custo.

1.2 Objetivos

Este trabalho tem como objetivo prover um mecanismo de roteamento interdomínio que explore os benefícios do paradigma SDN para resolver os problemas de inconsistências causadas pela visão fragmentada da rede presente e que possa evoluir facilmente de acordo com as crescentes demandas.

Adicionalmente, este trabalho tem os seguintes objetivos específicos:

1. Implementar uma aplicação com funcionamento semelhante ao do protocolo BGP utilizando os princípios do paradigma SDN.

2. Identificar quais características do BGP podem ser implementadas e quais não são necessárias na solução desenvolvida para SDN.

3. Prover uma solução que permita extensibilidade, ou seja, que possa incorporar modificações com facilidade; que possa evoluir.

4. Prover uma solução que permita abstração, ou seja, favoreça a criação e expressão de políticas de maneira mais fácil.

1.3 Contribuições

Enumeramos e descrevemos a seguir as principais contribuições deste trabalho:

1. Um estudo detalhado das causas dos principais problemas encontrados no roteamento interdomínio da Internet.

2. A apresentação de uma solução que possibilite inovação utilizando a abordagem SDN, mostrando como os problemas de inconsistência atuais podem ser resolvidos, bem como que mostre que a solução proposta é escalável e pode se recuperar de falhas no roteamento.

1.4 Estrutura do documento

Uma parte deste documento foi originalmente redigida na língua inglesa. Para atender às normas da Universidade Federal do Amazonas, dispusemos este primeiro capítulo como um resumo estendido na língua portuguesa e apresentaremos o restante desta dissertação em formas de apêndices, na língua inglesa. Este documento está estruturado da seguinte forma:

O apêndice B apresenta uma contextualização acerca do modelo de roteamento interdomínio utilizado atualmente na Internet, analisando suas principais características e identifica na literatura os principais problemas do roteamento interdomínio atual e suas principais causas.

O apêndice C apresenta uma revisão da literatura com os principais trabalhos que tentam resolver problemas da arquitetura atual da Internet e do roteamento interdomínio, provendo inovação. Analisamos a influência e contribuição que eles exerceram em nosso trabalho.

O apêndice D descreve nossa solução, o componente de roteamento interdomínio Inter-SDN, detalhando sua arquitetura, explicando seu funcionamento, identificando as características semelhantes ao BGP e como pode vir a resolver os problemas do modelo de roteamento interdomínio atual.

O apêndice E os experimentos realizados, as métricas utilizadas e os resultados obtidos.

O apêndice F apresenta as considerações finais e os trabalhos futuros a partir do trabalho apresentado nesta dissertação.

Apêndice A

Introduction

A.1 Motivation

The current Internet is partitioned into network groups under administrative domains called Autonomous Systems (ASes). Communication and reachability of information across different ASes is performed by the Border Gateway Protocol (BGP) (Rekhter et al., 2006), which has been widely deployed for many years and serves as the *de facto* Interdomain routing protocol of the Internet.

However, the structure imposed on the Internet by its adoption yields an architecture which lacks flexibility to allow and foster innovation (Raghavan et al., 2012), (Feamster et al., 2004), (Koponen et al., 2011). Routing inconsistencies and anomalies, policy conflicts, delayed convergence, and security inefficiency against DDoS attacks are, among others, some issues that the current architecture does not address.

Several solutions have been proposed to solve such problems, but the acceptance and deployment of these solutions are a difficult task. Besides the challenge to overcome BGP's rigid structure, a global agreement among ASes is required to deploy such enhancements. This complexity makes the BGP behavior unpredictable and prone to errors. Such undesirable situation is mainly due to the fact that the Internet's architecture and infrastructure are tightly coupled. As a consequence, evolution of its various protocols strongly depends on network equipment used, and this dependency on a specific infrastructure and hardware creates a barrier to architectural innovation.

An architecture that enables innovation must be extensible and abstract (Koponen et al., 2011). An architecture is extensible when it permits that new funcionalities or applications be easily added or replaced, evolving according to new needs. An architecture is abstract when low-level details are not considered, making the creation of applications faster and trustful. Any significant change on network architecture involves considerable costs to network vendors and operators.

Many BGP issues are originated in the exchange of interdomain routing information between routers inside an Autonomous System, task performed by the communication between BGP routers inside the same domain, known as iBGP (internal BGP). Although iBGP provides scalability, it introduces anomalies on routing and forwarding, as black holes and loops, caused by the protocol oscillation (instability). Besides that, the ability of right information distribution that iBGP performs may be affected by the inclusion a solely iBGP session. Techniques to avoid such anomalies are intensively researched (Vissicchio et al., 2012).

With a logically centralized architecture, with the routing decisions separated from routers, many of these problems could be solved. A Routing Control Platform (RCP) placed on each AS, logically centralized, with a full view of the whole domain, and with the routers only performing the forwarding task, could be more easily configurable, manageable and less prone to errors (Feamster et al., 2004).

A recent approach that provides the separation between the network infrastructure and the architecture, creating new possibilities, is the Software Defined Networking approach (SDN) (McKeown et al., 2008). SDN effectively separates the network on data plane and control plane, and is currently used on several datacenters. Although the SDN and other recent approaches have potential to enable the separation between infrastructure and architecture, enabling the architecture innovation, little solutions has been made to use this approach to address current interdomain issues.

We argue that exploring interdomain routing mechanisms within a non-rigid distributed architecture is more likely to overcome those issues than imposing any evolutionary modifications on the current Internet. Furthermore, such approach may evolve according to demands while supporting innovation. In this work we present, as a proof of concept of this idea, the results of our experiments using an interdomain routing mechanism within a set of SDN domains. Since SDN enables innovation through the development of high-level applications with abstraction and extensibility for enterprise networks, we extend the SDN paradigm beyond its primary target application by adding an interdomain routing component. In our proposed solution, an Autonomous System, or Domain, is defined as an *SDN Domain* with its own controller running a dedicated application to route packets outside its limits. We call such application the *Inter-SDN Routing Component*. We advocate that this approach makes the tasks of managing, troubleshooting, and designing secure networks easier.

A.2 Objectives

This work objective is to provide a mechanism of interdomain routing that explore the benefits of the SDN paradigm to solve the issues of the current approach, and that it may evolve easily according to demands.

Additionally, this work has the following specific objectives:

1. Implement an application with a behavior similar to BGP, but in a SDN archtecture.

2. Identify which BGP characteristics may be implemented and which are not necessary in the solution developed for SDN.

3. Provides a solution that enables the extensibility and abstraction, favoring the policy creation and expression.

A.3 Contributions

We enumerate and describe the main contributions of this work, as follows:

1. A detailed study on the causes of the main issues found on Internet interdomain routing.

2. The presentation of an innovative solution using the SDN approach, showing how the current issues may be addressed, as well as that the proposed solution is scalable and may recover from failures on routing.

A.4 Document Outline

Appendix B presents a background about the interdomain routing model currently used on Internet, analysing its main features, and identifies on the literature the main issues on current interdomain routing and its main causes.

Appendix C presents a literature review with the main works trying to solve problems of the Internet architecture and the interdomain routing, providing innovation. We analysed the influence and contribution that they exercised on our work.

Appendix D describes our solution, the Inter-SDN routing component, detailing its architecture, explaining its working, identifying the features similar to BGP and how may it solve the problems of the current interdomain routing model.

Appendix E present the experiments, used metrics and obtained results.

Appendix F presents the final considerations and the future work.

Apêndice B

Background on Interdomain Routing

B.1 The basics on Interdomain Routing

The ASes that compose the Internet are managed by network owners and administrators (Medhi and Ramasamy, 2007; Halabi and Mcpherson, 2000). Each AS executes, according to its policies and topologies, one or more protocols to provide routing within its borders. These protocols are called Interior Gateway Protocols (IGPs). The set of IGPs in one AS runs independently of the IGPs in other ASes. Figure B.1 depicts some ASes connected.

To handle the unavoidable increase of forwarding tables required for these operations, domains with common policies were agreggated into ASes so that the Internet could scale in a structured fashion. As a consequence this created the need to enable communication among ASes connected worldwide through well defined "routing" policies. The Exterior Gateway Protocol (EGP)[1] is a standard forwarding mechanism that allows communication among ASes and which has two well known approaches. One

[1] This is not the earlier EGP of the core ARPAnet model.

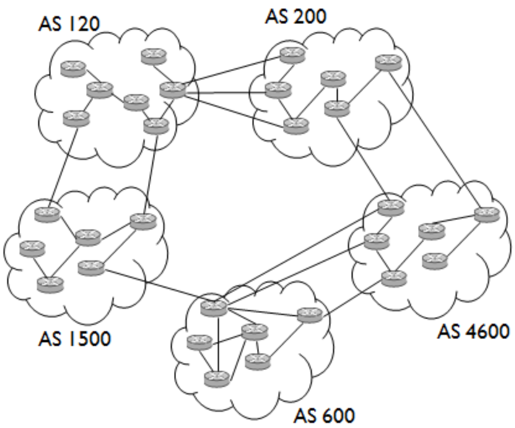


Figura B.1: Autonomous Systems model on Internet

such approach is the EGP protocol proposed in (Medhi and Ramasamy, 2007; Rekhter et al., 2006) which assumes that there is only one path between any two different ASes. The other one is BGP—the most widely used EGP protocol. Figure B.2 depicts the basic interdomain routing organization.

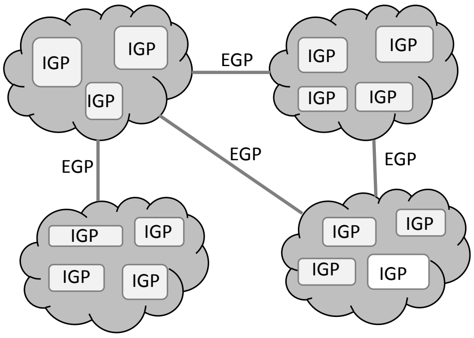


Figura B.2: Routing architecture on current Internet

As mentioned before, each AS is viewed by the others as a single administration entity, but it need an unique identifier. This identifier is provided by an Internet Registry or service provider, under responsibility of organizations divided around the world by regions. The identifiers comprise IP addresses and Autonomous System Numbers (ASNs).

On Internet the ASes relationships is modeled by contracts in exchange of traffic, and an hierarchy exists between the ASes. Each domain has its own interests expressed on its policies. This is the reason why the routing policies expressiveness must receive so many importance on interdomain routing study. Each AS might be defined as *customer, client* or *peer* on the relationship with other AS. In the relationship between customer and provider, the customer pays the provider to has access to Internet. A provider may be a customer of a larger AS. Two ASes are peers when they exchange traffic without payment, for benefit of both ASes (Gao, 2000).

B.2 Current Internet Interdomain Protocol: BGP-4

The Border Gateway Protocol (BGP) is a path vector protocol that has the important task of connecting the ASes in the whole world, making the Internet possible. It enables the exchange of information among the ASes, so that one can reach each other (Medhi and Ramasamy, 2007). The current BGP version is 4, described in (Rekhter et al., 2006).

The communication between ASes takes place with the creation of TCP-like connections, rather than using a link state protocol. While the latter does not scale well for an Internet with a rapidly increasing number of ASes, the former does offer reliable characteristics that allow a BGP implementation by means of the path vector protocol. The distance between two ASes is measured in terms of AS hops, which is the number of ASes in the path between them. Generally, BGP chooses the shortest AS path when required to select a route, but sometimes a longer path is a better choice and it is selected. BGP sessions may be divided into iBGP and eBGP. The eBGP sessions are the BGP connections created between ASes. The iBGP sessions are responsible for distributing the eBGP learned routes to the BGP routers within a domain (Medhi and Ramasamy, 2007; Rekhter et al., 2006).

There are four types of BGP messages: OPEN, UPDATE, NOTIFICATION and KEEPALIVE (Halabi and Mcpherson, 2000; Medhi and Ramasamy, 2007; Rekhter et al., 2006; Zhang and Bartell, 2004). The OPEN message is the first message exchanged, and contains information about the protocol version and the AS number, among others. The UPDATE message transports route information. It is used to announce new routes on network as well as to withdrawal the unavailable ones. The KEEPALIVE message is sent by each AS periodically to check if the connection persists. The NOTIFICATION message is used to inform about an error on message exchange. This message closes the defective connection between the BGP peers.

AS routers running BGP are called BGP speakers. Each BGP speaker has a Routing Information Base (RIB) that is divided into three parts: Adj-RIBs-In, Adj-RIBs-Out, and Loc-RIB. The Adj-RIBs-In store the routing information received by BGP peers. With such a base it is possible that a single prefix has several paths. The Loc-RIB stores the routes used by an AS to reach destinations on other domains. In a Loc-RIB there is only one prefix for each prefix stored, meaning that repetition of prefixes is not allowed. The Adj-RIBs-Out store the routing information, allowed by domain policies, to be sent to neighboring ASes through UPDATE messages (Medhi and Ramasamy, 2007)..

BGP has some attributes called path attributes, that describes the characteristics of the path information stored on BGP router RIBs. These attributes are crucial in the routing selection process. The BGP path attributes are: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, LOCAL_PREF, ATOMIC_AGGREGATE, AGGREGATOR, COMMUNITY, ROUTER_ID. These are the main path attributes, but may exist others, depending on the BGP implementation on the AS (Zhang and Bartell, 2004). Our solution presents some attributes based on BGP path attributes, but some are not present and new ones was created for the SDN approach, which will be detailed on next sections.

To determine the best path to a route- a process called path selection process- a BGP speaker must had first ran an import policy and maybe some filters on the routes received through UPDATE message. A route may be a new one, a withdrawn route or a replacement route. The Local_pref attribute may be used to ease the choose of a route. After that, a degree of preference is set to the route. The best route is recorded in Loc-RIB. The BGP selection path algorithm works as follows: First, the import policy and the filter checks if the IP prefix should be accepted or discarded. If accepted, the highest Local_pref is applied. If there is more than one route to the same IP prefix, the route originated from one BGP router, from inside the domain, is preferred. If more than one route persist, the one with the shortest AS_Path attribute should be chosen. The next tie-breaking rule is the ORIGIN attribute; the lowest value is chosen, beginning with the value of IGP, EGP and Incomplete, as last value of ORIGIN. If a draw persists in route selection, the MULTI-EXIT-DISCRIMINATOR attribute is used, being selected the route with the lowest value. If the search continues unresolved, route originated from EBGP over IBGP is preferred. The next tie-breaking rule is the route with lowest cost to Next-Hop. Next, the route received from neighbour router via EBGP with lowest BGP identifier is chosen. In last case, the route learned via IBGP from neighbour with lowest identifier is the selected (Medhi and Ramasamy, 2007).

To introduce new features on BGP there is an optional parameter, the *Capabilities Negotiation*. With this parameter, a domain may specify additional resources in the OPEN message and sends it to its peer. If the BGP peer does not support the capability negotiation, a NOTIFICATION message is received, with the error subcode "Unsupported Optional Parameter". This optional parameter has the advantage that the connection is not permanently closed, and the BGP speaker are encourage to try to re-establish the connection without the Capabilities parameter in the OPEN message optional field (Halabi and Mcpherson, 2000).

Other requirements of an interdomain routing are extensibility and flexibility on routing policy expressiveness (Zhang and Bartell, 2004). Even though BGP is a widely adopted standard that makes the Internet interdomain routing possible, yet as discussed below, there exist many issues to be solved due to its rigid structure.

B.3 Issues on Interdomain Routing

The Internet has evolved in many aspects in the last decades, but for its architecture. A number of modifications have been deployed to fix some protocols and services' issues, but problems hard to tackle, due to the network's rigid architecture, remain unsolved. This creates a barrier to innovation, causing an Internet "ossification". There have been many efforts to overcome such rigidity in order to make the Internet more secure, with better support for mobility, and more data-oriented. However, the lack of architectural modularity complicates the deployment of such solutions. The desired innovation on the Internet will happen when it becomes possible to modify intended system applications without being required to change others, and also without the need for a strong level of global coordination (Koponen et al., 2011).

The architecture is the Internet's layer which runs the protocols that deal with packet traffic and which also requires global agreement. The infrastructure, on the other hand, refers to the network equipment like routers, hubs, switches, and cables. The infrastructure and architecture of the current Internet are strongly coupled, meaning that any changes to protocols, e.g. the IP protocol, would require changing all network equipment that utilizes them. This is a great barrier to architectural innovation (Raghavan et al., 2012). Services as QoS are being even more required by ISPs clients, but BGP does not present these features because it was originally concept to send and receive route information messages. (Raghavan et al., 2012)

Furthermore, the interdomain routing mechanism itself presents crucial problems. One of them is its inefficiency in responding to the growth in demand. This happens, in part, due to the fully distributed path selection model used by ASes in which multiple routers within a domain are embodied with control and data planes. Distribution is necessary to achieve scalability, but many configuration states are not properly replicated under this model. Nowadays, implementation and deployment of network policies require dealing with the configuration of several routers. Moreover, the routers residing in a domain depend on each other's configuration, but they have no knowledge of each other's current state. As a consequence, forwarding loops and oscillations may occur. Separating the routing process from physical routers could help address these problems (Feamster et al., 2004).

BGP is a complex protocol mainly because of new features that have been added to attend the demand for more flexibility and scalability. This yields unpredictable behavior making error prevention a hard task to tackle. Another problem is related to the network routing components called *route reflectors*. A small network may use a full-mesh topology of iBGP sessions to ensure a consistent state of RIBs inside its domain, but this approach is not scalable for larger networks. Route reflectors were created to address this scalability problem; they are responsible for making routing decisions consistent using IGP path costs as their metric. The issue that arises is that an iBGP route reflector may select and distribute a route different from the one chosen by the other routers in the domain, providing an AS with distinct routes to the same prefix, which leads to protocol inconsistencies, e.g. forwarding loops.

Forwarding loops are not the only anomalies that may be caused by an incorrect interaction between iBGP and IGP within an AS; black holes, for example, may also occur. The main challenge in avoiding such inconsistencies is to ensure *dissemination correctness* (Griffin and Wilfong, 2002), a property that guarantees the non-existence of anomalies caused by route propagation in iBGP. However, verification and management of this property is, in practice, computationally intractable (Vissicchio et al., 2012) because the iBGP-IGP relation is highly distributed within a domain.

BGP also presents security problems, such as the lack of an authentication mechanism. Any IP prefix may be advertised by a BGP router and the message contents might be completely changed by routers' filters (Yannuzzi et al., 2005). Such a freedom exposes the architecture to attacks, and even misconfiguration can generate inconsistencies.

The fast growth of BGP routing tables has a direct impact on the protocol scalability. This is mainly caused by the increasing number of stub ASes that become multihomed in order to improve their connectivity, resilience, and load balance (Yannuzzi et al., 2005). Regarding policies, some tasks are difficult to be expressed using BGP mechanisms. For example, to change the traffic flow from one path to another is not a trivial task because it involves the identification of a subset of prefixes to carry traffic, modification of import policies, and change of path attributes (Feamster et al., 2004). Conflicts may occur because of the freedom each AS has to create and deploy its own routing policies without a global coordination involving the related domains (Yannuzzi et al., 2005). Finding the right balance between adequate expressiveness, one that preserves the policy autonomy of each AS, and the coexistence of different, conflict-free routing policies is a challenge for interdomain routing.

On what concerns to policies, some tasks are difficult to be expressed using BGP mechanisms, for example, make traffic flows from one path to another, is not a trivial task, which involves identify subset of prefixes that carry the traffic, modify import policies and change path attributes (Feamster et al., 2004). More expressiveness would be necessary to create policies that could do it easier, in a higher level manner. But provides great expressiveness and guarantee routing convergence is also a challenge. Each AS is free to create and deploy its routing policies; but conflicts may occur, since there is absence of a global coordination between the domains relationship (Yannuzzi et al., 2005). The lack of global coordination on routing policy makes traffic engineering on BGP very complex. The balance of good expressiveness, preserving the policy autonomy of each AS, and the coexistence of different routing policies without conflicts, is a challenge of interdomain routing.

One alternative to address problems generated by policy conflicts is the use of the Internet Routing Registry (IRR) that stores routing policies described on a specific language, e.g RPSL (Alaettinoglu et al., 1997). But this approach also has some problems. The registry may not be updated by ASes that do not want to publish their policies on the IRR. Moreover, even if all policies were registered on the IRR, the problem of convergence detection—due to checking all recorded registries—is NP-Complete. Another issue is that IRR may guarantee convergence under certain topologies, but when failures occur BGP might still not converge (Gao, 2000).

Convergence on interdomain routing is the process of updating all BGP routing tables to the synchronized state. Slow convergence must be avoided because it introduces inconsistencies that may negatively affect the Internet (Zhang and Bartell, 2004). When the convergence delay is relatively high, a source may end up not being notified when a destination becomes unreachable, leading to packet loss (Zhang, 2004). The BGP convergence time is slow, in the range of tens of seconds (Yannuzzi et al., 2005). Besides that BGP is very sensitive to minor changes on its parameters which directly impact convergence time (Bremler-Barr et al., 2003).

The distributed path selection used in a domain prevents the AS from having a global view and hinders the control of traffic on its networks. The large and growing number of prefixes on Internet also makes the convergence slow and keep the scalability is a challenge. Today there are more than 320,000 IP prefixes on Internet BGP routing table. The distributed path selection used inside domain, as mentioned before, prevents the AS of a global view and control of the traffic on its networks. Besides the slow convergence, this partial view of the AS may lead to suboptimal path choices, non-deterministic and, sometimes, prone-to-error behavior.

When referring to convergence, is important to mention one BGP timer called MRAI

(*Minimum Route Announcement Interval*), which is the time the BGP speaker should hold before it advertises new route information to its peers. It is necessary because if each BGP speaker advertise every new update on its table, there would be an impractical number of BGP messages being exchanged at a time, flooding the Internet with routing table states changing constantly, what would delay the convergence to an update state. With non-small value of MRAI set, the BGP speaker waits a certain amount of time before send a new advertisement. But if MRAI value be high, the BGP speakers will take long to notice about network updates, what would cause also a slow convergence. Therefore, the MRAI parameter should be set with a value that minimizes convergence delay, which is a complex task, once this best value is dynamic and hard to predict (Oprescu et al., 2010).

On Internet, the MRAI value is generally of 30 seconds. The optimal MRAI value actually depends on the topology and the type of experiment executed on the network (Yannuzzi et al., 2005). The absence of a global view of the domain is one of the factors that makes it difficult to predict the best value for MRAI.

Apêndice C

Innovation on Internet Architecture

Many of the issues discussed above are difficult to be addressed because the Internet architecture was not designed to support a high level of modularity and innovation. The Internet was not design to support a high level of modularity and innovation, the reason why its architecture remains relatively unchanged for years. Clean redesigns were proposed to meet future requirements, but these approaches do not promote incremental changes and without global agreement their deployment is unfeasible. An Internet with architectural components that incrementally evolve is necessary. To allow incremental changes it is necessary a component-based architecture rather than a layer division-based modularity as exists on the current Internet. A truly modular architecture should also offer extensibility and abstraction properties (Koponen et al., 2011). Interdomain Routing is one of the architectural components that should evolve without requiring the modification of some aspects of the physical infrastructure.

In the following sections we present some efforts being pursued that seek to allow innovation on Interdomain Routing for a modular Internet architecture. SDN proposes a paradigm shift on how networks are structured and managed which supports an actual modular architecture. Exploring SDN features, Future Internet Innovation (FII) is a proposal that attempts to achieve a modular Internet architecture. The Routing

Control Platform (RCP) and pathlets deal directly with interdomain routing, proposing innovative solutions and attempting to address many issues of BGP; solutions that may be utilized by FII and similar frameworks. Recently, the AgNOS framework has extended SDN beyond enterprise networks by means of software components for interdomain routing applications.

C.1 Software-Defined Networking

The SDN approach has the potential to bring innovation to the Internet because it enables architecture diversity and network modularity. It divides a network in Control and Data planes, each one with a well defined function. In SDN a network is managed through a logically centralized controller that rules a group of switches using a standard interface. These controlled switches may be from different vendors (Drutskoy et al., 2013).

The Data Plane comprises the network physical elements responsible for traffic forwarding like switches, hubs, and routers. The Control Plane is composed by the Network Operating System (NOS), the NOS Interface, and the applications that run on top of it. NOS is responsible for managing the network resources and enforcing the actions decided by the applications onto the Data Plane elements. The communication between the Data and Control planes is done by the standard interface which must be deployed on network equipment. The most well known and used interface for the Data and Control planes' communication is the OpenFlow (OF) protocol (McKeown et al., 2008) which provides APIs to install packet-forwarding rules, store and query traffic statistics, and learn topology changes (Drutskoy et al., 2013). With SDN, developers create applications without worring about low-level details. The API provided enables fast prototyping, implementation, and deployment of new applications for managing the networks in a fast and less prone-to-error manner.

Figure C.1 depicts the SDN basic architecture. The physical network equipment, including hosts and OF switches, reside on the Data Plane. The Control Plane is comprised by the Controller (NOS) and a set of applications that manage the network.

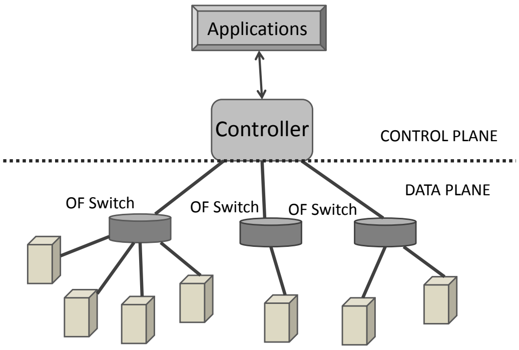


Figura C.1: SDN architecture

C.2 Framework for Internet Innovation

The Framework for Internet Innovation (FII), proposed in (Koponen et al., 2011), is an effort to enable the evolution of the Internet architecture and support diversity, i.e. coexistence of different architectures. Many issues related to architectural problems on the Internet have solutions that have not yet been deployed because of the existing barrier to innovation; however, as FII supports different architectures, these solutions could be deployed on FII to address these issues. FII authors do not attempt to develop a set of components to solve the aforementioned problems, but propose a framework which enables the coexistence and cooperation of components created by others. The less specified and constrained by FII the network is, the more freedom there will be to innovate.

Three features are identified as fundamental parts of a framework. The first is an interface to enable communication among domains, i.e. a mechanism for interdomain routing. The second is an interface between network applications, i.e. a network API.

The third is an interface to provide security against attacks, such as Denial of Service (DoS) attacks.

With Network API, FII may use different name schemes, requiring only that each used namespace be specified in a standard manner. With this feature, it is possible to have a network using the traditional IP addressing scheme communicating with a network that uses a different address scheme, i.e., a different namespace. To mitigate DoS attack, FII uses a mechanism called *Shut-up Message* (SUM), that allows the victim to send a message to the sender to stop it sending packets to this destination.

Some architectural components are difficult to change and innovate upon. Such components are called *anchors*. One of the main anchors the FII authors identify is the interdomain routing, because deploying any modification on it requires global agreement. Besides, it is necessary the elaboration of documents that standardize the way the protocol modifications must be implemented in order to keep the Internet working, and this process may take years. As the FII must be built using the SDN approach, we propose our Inter-SDN component as a candidate capable of meeting FII interdomain routing requirements.

C.3 Pathlet Routing

Pathlet Routing (Godfrey et al., 2009) is a protocol that provides interdomain routing with highly flexible policy support. The protocol consists of a pair of vnodes and pathlets. The vnodes are virtual nodes created by ASes according to their policies. Pathlets are fragments of paths represented by a sequence of vnodes. With the pathlet routing approach the Internet might be represented by a virtual network, independent of the physical topology or infrastructure. The number of vnodes used is decided by the AS, making the task of declare services and policies constraints easier to network owners. Pathlet routing implements two services: multipath routing and source routing.

Source routing allows the sender to decide which route the packets may take to reach their destination. Multipath routing provides the possibility of having multiple path options during the routing with the source selecting one path that may quickly change in case of failures. Pathlets enable simulation of many interdomain routing protocols, including BGP, and present a new kind of policy, called Local Transit (LT) policy. With LT policies, the ASes policies have local constraints with FIBs scaling on the number of neighbors and presenting a small size—when compared to FIBs of routers that use path vector policies—as BGP does.

This approach has, however, the disadvantage of constantly changing the header of the packets that carry the routing information. This happens because senders concatenate the selected routes, represented by a list of identifiers, in the packet header. It means that packet size also changes during the packet transit between ASes, which is a critical problem because packet size becomes unpredictable.

C.4 Routing Control Platform

The Routing Control Platform (RCP) is an approach proposed to separate the routing process from the routers, allocating it on dedicated servers with a complete view of the network. With RCP, many iBGP issues are avoided, replacing route reflectors, and avoiding inconsistencies (Feamster et al., 2004).

RCP is logically-centralized on each domain. Besides the consistent state obtained by the complete knowledge of the domain, the RCP architecture provides easy maintenance since any changes on the interdomain routing protocol require dealing only with the RCP server.

RCP attempts to provide the benefits of consistency present on the full-mesh iBGP configuration and the scalability that route reflectors allow. However, differently from route reflectors, RCP may send a distinct route to each router in a domain. RCP comprises of three elements: an IGP viewer, a BGP Engine and a Route Control Server (RCS). RCP needs information about topology and best path to destination from each router; this information is obtained from the IGP viewer module. The BGP Engine allows the RCP to "tell"the routers about the routing decisions taken. Besides that, BGP Engine is responsible to make RCS learn new BGP routes. The RCS computes the information of topology and BGP routes available to compute the best route for each router on domain (Caesar et al., 2005). RCP may be easily deployed on domains because it uses iBGP sessions for the communication between BGP speakers and the RCP server.

RCP's drawback is that it is constrained by the limitations of the protocol used; even though the complete network view is provided to avoid most inconsistencies. This means that RCP may not evolve unless the BGP protocol also changes, which creates an obstacle to innovation.

C.5 AgNOS

The Agent for Network Operating System (AgNOS) (Passito, 2012) is a framework based on multiagent systems built on top of the abstractions provided by SDN networks. AgNOS enables the autonomous control of the SDN network, assuming that the Future Internet will be organized on domains composed by autonomous networks.

Multiagent systems are composed by agents that interact with each other to achieve established goals, cooperating and negotiating when different interests between domain interactions raise conflicts. The multiagent knowledge base should represent services, protocols and components, and other elements of the network.

An autonomous network should be capable of receiving high level instructions and configuring itself, detecting and troubleshooting problems, and adapting to changes on the network. Autonomous networks use agents to enable the creation of components that help the networks manage their own behavior. One important case study presented by (Passito, 2012) is networking traffic management between domains using an interdomain routing application—the Inter-AS component (Bennesby et al., 2012). In this application, the Inter-AS component does the routing, but AgNOS agents perform the tasks of routing negotiation and cooperation when failures occur.

One important case study presented by (Passito, 2012) is networking traffic management between domains using an interdomain routing application—the Inter-AS component (Bennesby et al., 2012). Figure C.2 presents the AgNOS framework in an inter-AS setup.

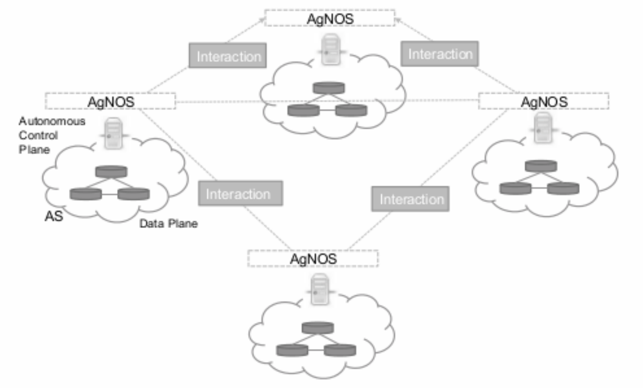


Figura C.2: The vision of AgNOS in an inter-AS setup

In this application, the Inter-AS component does the routing, but AgNOS agents perform the tasks of routing negotiation and cooperation when failures occur.

Apêndice D

The Inter-SDN Component

This appendix presents the Inter-SDN routing component, describing its architecture, its core elements, and its behavior. Besides that, there is a discussion of how may Inter-SDN solve current Interdomain routing issues. The experiments and its results are also described on this appendix.

D.1 Architecture Description

The Inter-SDN Routing Component is responsible for interdomain routing on SDN networks, similarly to the work performed by the BGP protocol in the current Internet. To make interdomain routing possible on networks using the SDN approach, the Inter-SDN Component co-works with other components. This is the advantage of the innovation provided by modularization in the SDN approach. One may easily create different applications that may work together, each possibly being responsible for a specific network task.

The Inter-SDN Routing Component is the implementation of a routing protocol based on BGP's most important features, but with some architectural differences to meet SDN principles. This component extends the Inter-AS component (Bennesby et al., 2012), which implements existing features of the Internet's current interdomain routing, taking advantage of SDN abstractions. This allowed an easy, fast, and less prone-to-error implementation of Inter-AS. Furthermore, the new component can be easily extensible by adding new features, as well as tested and deployed in a much faster and easier way than it is done today in the current interdomain routing. We could have implemented a component with features completely different from those offered by BGP, but our goal was to show that a solution offering most of BGP's features implemented on a different architecture may solve many unsolved problems.

The Inter-SDN component needs to interact with a component providing intra-domain routing and with another enabling the communication between the domains. Additional components may be used to improve the domain functionalities, for example, a component that detects DDoS attacks (Braga et al., 2010) or that deals with traffic load balance within a domain.

Figure D.1 presents the structure of a Software-Defined Network and the elements necessary to have interdomain routing working properly. Observing the picture in a bottom-up order, we first see the dataplane. In our solution, we need at least three applications to make interdomain routing possible: one to perform the IGP task, another one that works as an EGP, and a third one responsible for the connection between EGPs from different ASes.

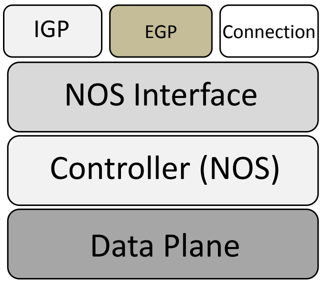


Figura D.1: SDN Domain Architecture

We chose to adopt the open source version of the Network OS NOX (Gude et al., 2008) because it was one of the first controllers created for SDN. Our solution uses OpenFlow (McKeown et al., 2008) as the protocol that makes the Data Plane understand the actions sent by the applications on the Control Plane. In NOX, applications are called components and the interface provided to support the components is the programmatic interface. The IGP protocol may be any component able to route packets inside the domain. A NOX component called Switch is used. The EGP protocol is our proposed solution, the *Inter-SDN Domain Routing Component*. The component responsible for the connection, called Messenger, opens a TCP connection with the AS peer and sends and receives messages composed by the Inter-SDN component. We could have implemented the Messenger functionalities in Inter-SDN reducing the number of components, but one of the SDN advantages is its well-defined modularity which eases code implementation and maintenance.

We consider that the Internet will continue being organized in domains. But in the Future Internet architecture, these domains will be managed by networks which deploy the SDN paradigm and will be named as SDN Domains.

A SDN Domain is the composition of one or more SDNs managed by the same set of application policies. Each SDN Domain has autonomy in relation to other SDN Domains, similarly to current Autonomous Systems.

Our solution emulates BGP main features because it aims to provide inter-communication between SDN domains. Interdomain communication performed by BGP has been responsible for the Internet technological success (Elmokashfi et al., 2012), and that's why BGP has been intensely studied over the last decades. We present next the main BGP features implemented on Inter-SDN and discuss what changed from the current protocol to our solution.

D.2 Component Description

The Inter-SDN component was implemented in C++ and currently runs on top of NOX. The communication between SDN Domains is made by the Messenger component that creates TCP connections on port 2603. This communication is similar to the sessions created by BGP routers to exchange reachability information. The messages generated by the Inter-SDN component are made of strings that Messenger converts to streams of bytes and sends to the destination established by the TCP socket connection. Due to space limitations, we will not detail all implemented features, instead we will cite only the most important ones and focus on the differences to BGP.

D.2.1 Messages

In our approach, a component called Messenger creates a TCP socket connection with the peer using port 2603. The Inter-SDN messages are composed by strings that are converted by the Messenger to a stream of bytes and sent to the destination through the TCP socket connection established. When the messages are exchanged the peers analyze the attributes and decide whether the message should be accepted. Sometimes there are policy needs to be fulfilled and the peers have to negotiate to maintain their connection.

Similarly to BGP, the Inter-SDN component has four message types. Each message has a header of fixed size. The Inter-SDN message header is composed by the marker and type fields. The marker may be used to check synchronization between the connected peers. The type field indicates the type of the message. Table D.1 depicts the Header Message fields format.

Tabela D.1: Header Message Format

Marker	Type

The message types Inter-SDN component deals with the following:

1- *Open*: The Open message contains fields: version, myAS, hTime, identifier, optParCode, and optParValue. The version field is used to verify if the version of the component is compatible on both peers. The peers need to agree about the version used. If they do not agree, the connection is closed. The current BGP version is 4. The Inter-SDN version is 1. The myAS field is a unique number that identifies the AS. The AS number may range up to 65535, but some of these numbers are private and not available. The hTime field indicates the time, in seconds, that the peer will wait to receive an Update or Keepalive message. If the time expires, the connection is closed. The identifier indicates the IP address of the sender. The optParCode and the optParValue are optional parameters. They may be used if the peers agree during the negotiation. The optParCode uniquely identifies the optParValue, which indicates a value to the parameter recognized by the peer at the optParCode field. Table D.2 depicts the Open Message fields.

Tabela D.2: Open Message Format

version	myAS	hTime	identifier	optParCode	optParValue

2- *Update*: The Update message consists of the following fields: withdrawnRoutesLen, withdrawnRoutes, attributeLen, attCode, attValue, and NLRI. Field withdrawnRoutesLen provides the length of the withdrawnRoutes field. The withdrawnRoutes field indicates prefixes previously announced that are no longer available. In this field, prefixes are indicated by the IP address followed by the network mask. The attributeLen is the number of attributes used in the Update message. The attCode is composed by the attribute flags and the attribute code. The flags indicate whether the attribute is well-known mandatory, well-known discretionary, optional transitive, or optional nontransitive. The attCode indicates the attribute codes of the Update message. In Inter-SDN the flags have the same values as in BGP.

The mandatory Inter-SDN attributes are: AS_Path, Next_Hop, and Local_pref.

The Origin and Multi_Exit_Disc (MED) attributes that exist in BGP are not used in Inter-SDN. The AS_Path lists the AS_Numbers through which the routing information has passed. The Next_Hop indicates the IP address that should be used as a next hop to reach the destination. The Local_pref indicates the preference the AS has for one route over others of a certain prefix. The local preference value for a route is generally determined by the local policies configured by the domain administrator. The Origin attribute is used in BGP to differ whether the route was learned by a peer inside its AS domain or from another AS. Origin is not necessary in Inter-SDN because inside an AS domain the routes from other ASes are only learned by the controller running the component. The Multi_Exit_Disc (MED) attribute is also not present in Inter-SDN. MED is an optional attribute used to discriminate routes to a prefix coming from multiple entry points. It is required in the BGP decision process when the other path attributes have the same value. Figure D.2 shows a BGP router in AS 1 receiving more than one route announcement to the prefix 10.0.0.0/24 from different peers in AS 2. The BGP peers in AS 2 will set different MED values to each route announced. Then, the route with the lower MED attribute is preferred. This attribute is not present in Inter-SDN because there is no chance the route update to a certain prefix would come from multiple points of an AS. In Inter-SDN, the interdomain routing is not performed on borders; the controller running Inter-SDN decides which exit point (port) will be used and through which switch the message will be sent, as Figure D.3 suggests. The protocol only uses one route to a prefix and may only announce this route. Thus, this renders the MED attribute useless in Inter-SDN.

The optParValue attribute indicates the value associated to a mandatory or optional attribute, depending on its code. Table D.3 depicts the Update Message fields.

Tabela D.3: Update Message Format

wdRoutesLen	wdRoutes	attLen	attCode	attValue	NLRI

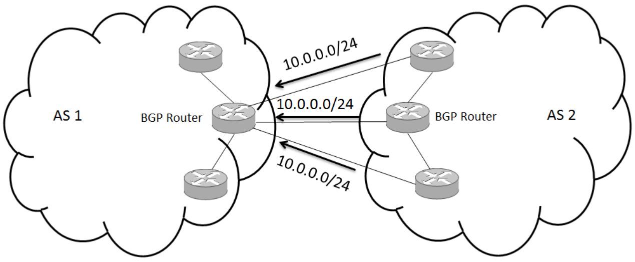


Figura D.2: BGP-based AS receiving routes for a prefix at multiple entry points

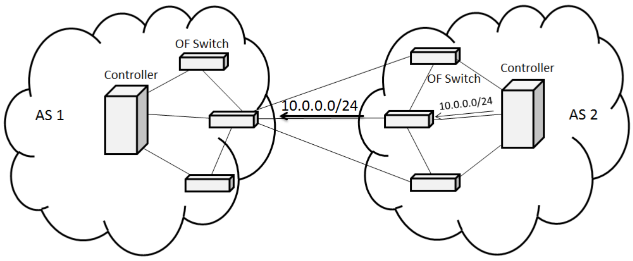


Figura D.3: SDN-based AS receiving a route for a prefix at one entry point

3- *Notification:* The Notification message consists of the following fields: errorCode, errorSubCode, and data. The errorCode field indicates the error type, errorSubCode provides details about the error, and the data field contains additional data information. Table D.4 depicts the Notification Message fields.

Tabela D.4: Notification Message Format

errorCode	errorSubCode	data

4- *Keepalive:* The Keepalive message consists of only a header message and a field with the socket ID, as shown in Table D.5. The Keepalive message fields inform which peer is associated to that connection and test whether the connection is still alive. If parameter holdTime expires and a Keepalive message does not arrive, a Notification message is sent, and the connection to the peer is closed.

Tabela D.5: Keepalive Message Format

Header	socketID

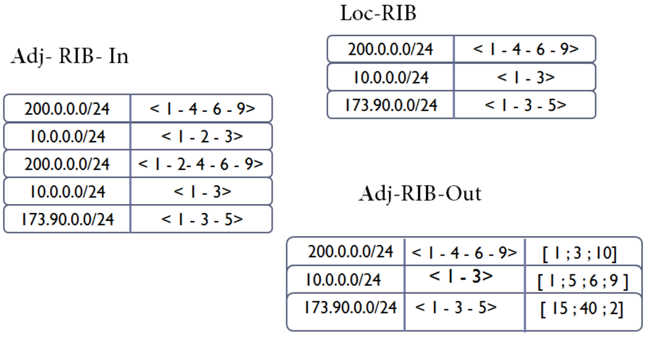
D.2.2 Information Base

The Routing Information Base (RIB) is used to store and advertise routing information. The main difference of RIB usage between BGP and Inter-SDN is that each BGP router in a domain has a RIB and needs to exchange Updates with its peers to maintain consistency from the AS point of view. In the Inter-SDN implementation, the RIB is not distributed in the domain routers, but logically centralized, running on the controller.

The RIB is divided in three parts: the Adj-RIB-In, Loc-RIB, and Adj-RIB-Out. The Adj-RIB-In stores the routes learned from peers. The routes located in Adj-RIB-In are queried when a route previously used by the AS is no longer available. The Loc-RIB stores the routes that the AS domain use. In Loc-RIB there is only one route for each prefix installed. The Adj-RIB-Out contains the routes that might be announced to the AS peers.

The Adj-RIB-In contains information not yet processed. When a new route arrives at the component, it checks whether the path has not been previously announced by the same peer. If the information is new under those conditions, it is stored in Adj-RIB-In. Input policies may be applied to choose the routes that may be used according to routing policies defined in the AS domain. When routing information is inserted in Adj-RIB-In, the decision process must be applied to verify whether the new route is better then the current route to that prefix installed in Loc-RIB. Then, the routes present in Loc-RIB might be advertised to the AS peers. Depending on the output policies the AS uses, some routes may not be announced to some of the AS peers.

Figure D.4 depicts an instance of the RIB, showing its three parts. In Adj-RIB-In all the routes are installed. The Loc-RIB contains only one route for each prefix which are accepted by the input policies. The Adj-RIB-Out contains the routes exported from the Loc-RIB. The last field of the table indicates the SDN Domains that may receive each advertisement according to the domain's output policies.



Adj- RIB- In

200.0.0.0/24	< 1 - 4 - 6 - 9>
10.0.0.0/24	< 1 - 2 - 3>
200.0.0.0/24	< 1 - 2- 4 - 6 - 9>
10.0.0.0/24	< 1 - 3>
173.90.0.0/24	< 1 - 3 - 5>

Loc-RIB

200.0.0.0/24	< 1 - 4 - 6 - 9>
10.0.0.0/24	< 1 - 3>
173.90.0.0/24	< 1 - 3 - 5>

Adj-RIB-Out

200.0.0.0/24	< 1 - 4 - 6 - 9>	[1 ; 3 ; 10]
10.0.0.0/24	< 1 - 3>	[1 ; 5 ; 6 ; 9]
173.90.0.0/24	< 1 - 3 - 5>	[15 ; 40 ; 2]

Figura D.4: RIB instance of Inter-SDN component

D.2.3 Route Selection Decision Process

The route selection decision process may be described as an algorithm that chooses a route to a destination when a previously announced route is no longer available, or a new route arrives at Adj-RIB-In. The BGP decision process has more steps than that of the Inter-SDN (Rekhter et al., 2006), because, with the change of architecture, some of the path attributes are no longer used on path decision, e.g. the MED attribute. A basic Inter-SDN[1] should use the following steps to select the best routes:

1. Local Preference

2. AS_Path

3. Socket Id

Local Preference is an attribute used to set the degree of preference among routes with the same destination prefix. If two routes have the same local preference value, the route with the smallest AS_Path is chosen. If the AS_Path lengths of the routes are the

[1] Note that more sohisticated versions of Inter-SDN can be easily added.

same, the last tie-breaking rule executed takes the route learned from the connection with the smaller Socket Id. When using the Messenger component to establish the TCP socket connection, the Inter-SDN creates a unique number associated to each connection. This number is the Socket Id.

D.3 SDN Domain behavior

The Inter-SDN Routing Component implemented for NOX works with the Switch and Messenger components. We are going to extend the proposed solution to other controllers. The whole system works as described in figure D.5.

The Switch component first creates the connections between the controller and OF switches on the network. The OF switches are comprised by a flow-table, a secure channel, the controller, and the OF protocol (McKeown et al., 2008). Instead of dealing with packets, i.e. sending them all to the controller, the OF switch only sends the first packet of a group that has the same header, and the following packets are treated in the same way. This approach makes scalability possible with NOX (Gude et al., 2008).

The figure D.5 depicts the system behavior process in eight steps, which we detail as follows. We consider that the SDN Domain has already exchanged routing information using Messenger and needs to deliver information from one host inside its SDN Domain to a host in other SDN Domain.

After creating connections inside the domain, the Inter-SDN calls Messenger to start a communication with neighboring SDN Domains and exchange routing information. When a host in one AS tries to reach another host located in another domain (1), the packets that reach the first OF switch have their headers compared with the entries of the flow-table. If no rule is found, the header of the first packet is sent to the controller (2), generating an event that is treated by the Switch component (3). The Switch component checks if the IP prefix belongs to its domain. When the destination is inside

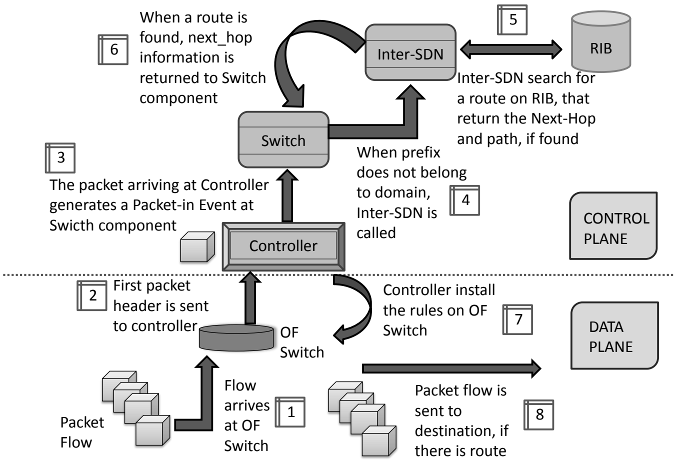

6
When a route is found, next_hop information is returned to Switch component
Inter-SDN
5
RIB
Inter-SDN search for a route on RIB, that return the Next-Hop and path, if found
Switch
3
The packet arriving at Controller generates a Packet-in Event at Swicth component
When prefix does not belong to domain, Inter-SDN is called
4
CONTROL PLANE
Controller
2
First packet header is sent to controller
OF Switch
Controller install the rules on OF Switch
7
DATA PLANE
Packet Flow
Flow arrives at OF Switch
1
Packet flow is sent to destination, if there is route
8

Figura D.5: Inter-SDN Domain Behaviour

the domain, the Switch component installs rules on the OF switches to take the proper action according to domain policies. When the destination address does not belong to the domain (4), the Inter-SDN component is called, taking the IP prefix from the Switch component and searching for it in RIB (5). When entries are found in RIB, information like the next hop AS and the number of hops to reach the destination are used to select a route. When a route is chosen, Inter-SDN returns to the Switch component the OF switch ID to send the flow and the outport (6). The controller installs the rules in appropriated OF switches (7) and the packet flow is sent to destination, if there is an available route (8). This process is repeated when the flow arrives at other domains, until it reaches the destination (Bennesby et al., 2012).

D.4 How may Inter-SDN solve interdomain issues?

The Inter-SDN component implements the main BGP features to provide interdomain routing while taking advantage of the benefits offered by SDN. Inter-SDN is prone to innovation because it takes advantage of architectural abstraction and extensibility properties. The abstraction is provided by the SDN paradigm, i.e. the API and NOS interface enable the creation of components without concern to low level details related to infrastructure. The developer does not need to know any details about particular vendor equipments. With this advantage new components may be rapidly prototyped and implemented, error correction is also easy, and there is no need to reinstall the solution in all the network switches. Instead, the updated software might be replaced in the controller, and the new rules may be installed automatically on the OF switch.

The component is extensible, i.e. features might be withdrawn and added easily, without complex standardization and wide global agreement. For example, our current Inter-SDN implementation does not provide source routing and multipath routing, as BGP. But unlike BGP, these features might be easily added to the code and deployed, without need of global agreement from all SDN Domains. The SDN Domains have autonomy to update their interdomain components and may negotiate their capabilities on message exchange. This innovation stimulates the creation of several applications for security, load-balance, and mobility, among others.

With Inter-SDN it is possible not to use Route Reflectors and solve most aforementioned inconsistencies problems. The SDN Domain is logically centralized and has a global view of the whole network. The intelligence and new decisions are made by the applications on the controller. With this approach, properties such as *dissemination correctness* (Vissicchio et al., 2012) are easier to guarantee. To solve security problems, e.g. DDoS attacks, using SDN some approaches were proposed (Braga et al., 2010) and tested on a framework (Passito, 2012) using a previous version of our component (Bennesby et al., 2012). Another important feature of an interdomain routing solution is the power of policy expressiveness. As SDN provides a complete view of the network domain and facilitates the development of applications to manage the network traffic, writing routing policies becomes a relatively simple task. Actions like dropping packets from one host, sending traffic to one specific port of certain switch on network, and rejecting traffic from one specific SDN Domain, among others, may be easily created and may compose the policies of a SDN Domain.

Although SDN is still in its early stages, its success, and the promising results we obtained point that SDN has a huge potential to be explored in the coming years, with emerging applications that we do not foresee with the current Internet architecture.

Apêndice E

Experimental Analysis

E.1 Emulation Environment: Mininet

In our experiments we used Mininet (Lantz et al., 2010), an environment developed to support large scale network emulation in a lightweight manner. Mininet is (a) highly flexible - allowing a wide range of types of topologies and new functionalities; (b) scalable - it is possible to have an environment with hundreds or thousands of switches in one laptop; and (c) realistic - it represents the true behavior of its physical counterpart achieving high confidence results when compared to real networks. Our experimental scenarios are composed of various instances of a simple, but significant, SDN Domain, and each one is formed by a NOX controller connected to a Mininet network containing one switch and two hosts.

E.2 Experiments reliability

For our experiments we made pilot studies to determine the number of experiment replications needed in order to obtain reliable experimental results (Garcia, 2008). The following method was used to determine the number of experiment replications required to obtain a reliability level of 95%.

$$n = \left(\frac{100 z \sigma}{r \overline{(x)}} \right)^2 \tag{E.1}$$

The r variable represents the desired accuracy of our experiment. As the degree of reliability is 95%, r value is 5%. The z variable represents the normal variate of the desired confidence level. The standard value of z for this degree of reliability is 1.96. The $\overline{x}$ variable represents the sample mean. The σ variable is the standard deviation, and may be calculated as follows:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left((xi) - \overline{(x)} \right)^2} \tag{E.2}$$

where N is the sample size and xi represents each value of the sample. Thus, we determined the number of replications n required to achieve the desired reliability for our experiments.

E.3 Emulation Results

E.3.1 First Experiment: Scalability

In the first experiment we measured the latency associated to the RIB growth between two SDN Domains exchanging data. This experiment provided good evidence about the solution scalability.

The RIB table of the source SDN Domain may influence in the latency. When the destination is in another SDN Domain, the source needs to look at its RIB to discover how to reach this destination, and to send the necessary information to the intra-AS component. The LocRIB is the first consulted to know how to reach a destination. In this experiment the best path to the destination is always already at the Loc-RIB—the

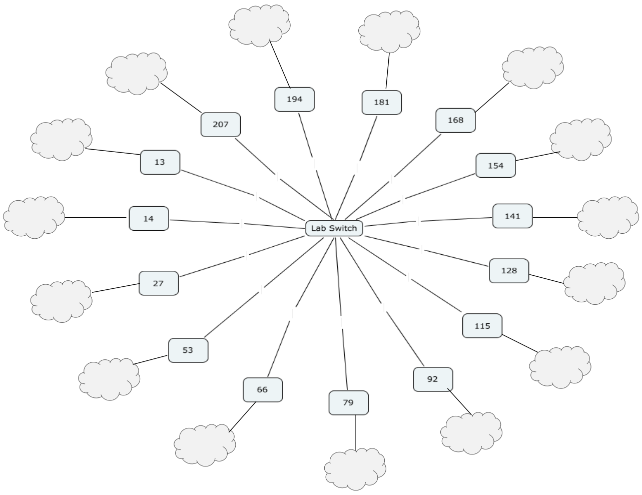


Figura E.1: Scenario used in experiment 1

decision process occurs when a new route to a prefix already at the Loc-RIB arrives, or when some prefix is removed from the Loc-RIB. In general, the more SDN Domains the entire network has, the bigger will be the RIB and the impact on performance and scalability of the solution.

We used different scenarios to analyse the performance of the communication between two domains with the Loc-RIB size increasing on each test. Figure E.1 depicts the scenario used for the first experiment. The numbered rectangles connected to the central rectangle labelled "Lab Switch"compose the backbone of the scenario. Each cloud in the picture comprises of 12 or 13 SDN Domains disposed on different topologies. There are overall 200 SDN Domains in the scenario.

We measured the response time when experimenting with 10, 50, 100, 150, and 200 domains. In the used scenario, there was a central group of domains connected, composing a backbone, each one connected to a group of several other domains. These groups of domains were designed using several different topologies, such as ring, linear, tree, star, and others. We assume that each SDN Domain has only one IP prefix. Since a Loc-RIB only has one route for each destination prefix, the Loc-RIB size of each

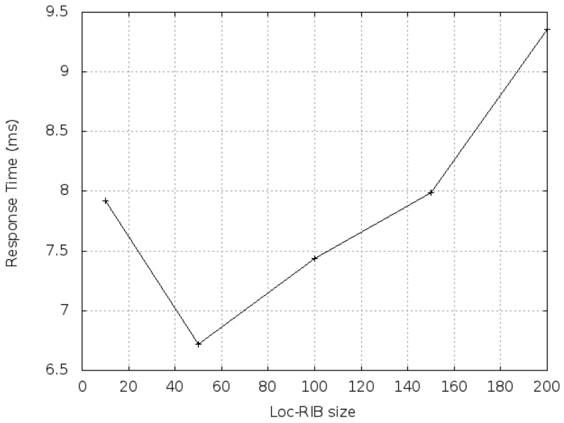


Figura E.2: Response Time variation with Loc-RIB growth

domain is equal to the number of the domains connected in the experiment. Figure E.2 indicates response time growth in relation to the Loc-RIB increasing size.

The RIB is implemented as a sequential search and, for each search, we forced the worst case to happen with the whole table being visited to find the information needed. The results demonstrate that there is little influence of the RIB size in the response time. This may be explained by the fact that in the SDN paradigm, mainly in the NOX/OpenFlow architecture used, not all the packets are sent to the network controller, but only the first packet of a flow. The delay only happens when a new flow needs to be installed on a switch. This makes the entire solution efficient. Thus, although the search is forced to operate in the worst case, the impact is very low, because the RIB search does not happen for all packets in the network. We see that from 10 up to 200 domains there is no significant difference, but the RIB size influence in the response time may be stronger in larger scenarios containing thousands of domains. The search might also be improved using other data structures, as b-trees or hash tables.

E.3.2 Second Experiment: Alternative path choice

The second experiment observed the behavior of the solution when a link is broken and an alternative path to reach the destination is necessary. If there is an alternative path in Adj-RIB-IN, it is placed in Loc-RIB and the new path becomes the choice to send the traffic. There may be a delay during the change to other destination, due to the processing of the decision process algorithm. We used three topologies to evaluate how much this convergence to a new path impacts the response time of the used scenarios, as depicted in Figure E.3. Each topology in figure has two communicating nodes emphasized in red. For each scenario, we measured the delay associated to 45 ICMP packet exchange. We measured the response time of each scenario without a link failure and with the link failure happening.

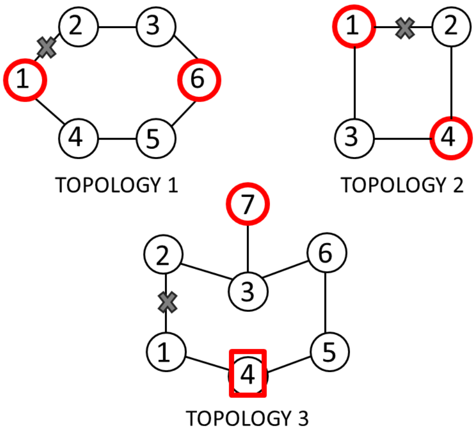


Figura E.3: Scenarios used in experiment 2

Topology 1 is comprised of six domains where domain 1 tries to exchange information with domain 6. The path chosen, beginning with domain 1, is 1-2-3-6. When the link between domains 1 and 2 is broken, domain 1 switches its path to 1-4-5-6. In this first scenario, although path 1-2-3-6 was the primary option, it was overloaded. When the new path started being used, the response time decreased. This result may be seen in figure E.4. The graphic E.4 shows that when the path becomes unavailable between packets 10 and 15 in the sequence of 45 packets there is a delay in the response time. The response time of the delayed response was of 55 ms.

Topology 2 depicts a scenario with four domains. Domain 1 exchange information

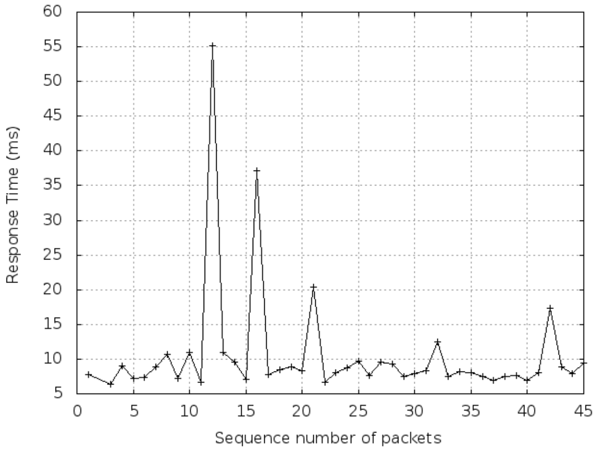


Figura E.4: Response Time variation with link failure

with domain 4 through domain 2. When the communication between domains 1 and 2 is broken, path 1-2-4 becomes unavailable, switching to path 1-3-4. The graphic E.5 shows that when the path becomes unavailable on packet sequence 15 in the sequence of 45 packets there is a delay in the response time. The response time of the delayed response was between 11 and 12 ms.

Topology 3 has seven domains. Domain 7 reaches domain 4 using path 7-3-2-1-4. When the link between 1 and 2 is broken, path 7-3-6-5-4 starts being used. With this scenario there is also a slightly higher response time. The graphic E.6 shows that when the path becomes unavailable between packets 25 and 30 in the sequence of 45 packets there is a delay in the response time. The response time of the delayed response was of 5 ms.

This makes evident that when the decision process is used, the convergence to a new path generates delay. Our experiments showed that the delay is dependent of the topology used and although the delay occurs, there was no packet loss during the new route convergence.

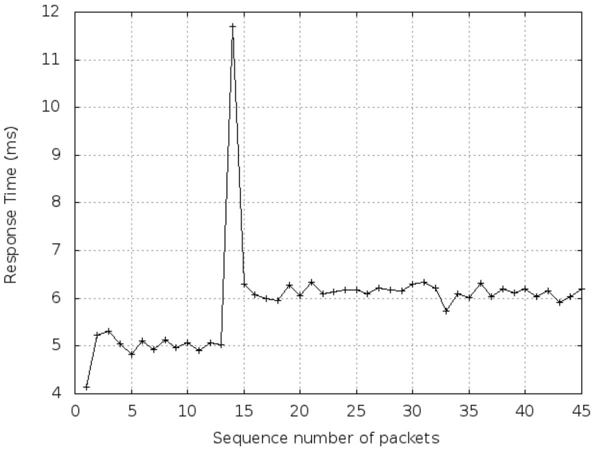


Figura E.5: Response Time variation with link failure

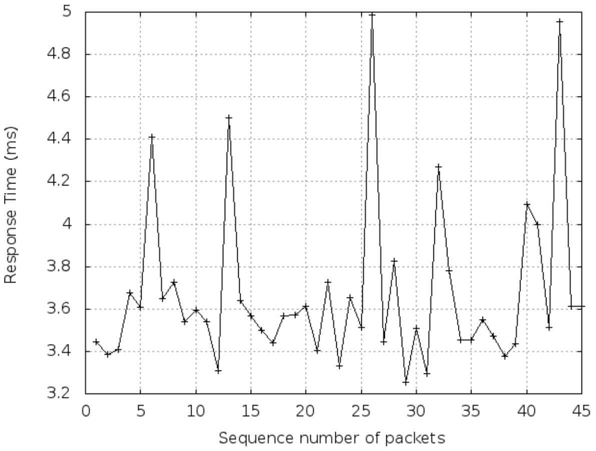


Figura E.6: Response Time variation with link failure

E.4 Additional Discussions

The SDN paradigm provides a logically centralized network control. One may ask about the feasibility of using SDN in a larger network domain, i.e. a domain containing multiple nodes on the network and which handles a large amount of traffic. The authors of (Raghavan et al., 2012) estimated that 10 Tbps is a good estimative for the bandwidth required for a large domain and 1500 cores would be enough to handle the traffic of a current big domain. This does not imply that it must exist only one controller by SDN Domain. More controllers might coexist and cooperate composing a logical control of the network. In this case, special care must be taken on controllers coordination to avoid inconsistencies.

Another issue is the single point of failure introduced by the centralized control model. If the controller goes down, the network may crash. We have done some parallel work that tackles this problem, providing resilience to SDN components using backup replicas (Fonseca et al., 2012). That solution works as follows: if a controller fails for some reason, another controller—previously used as a backup—assumes the network, seamlessly, with the same state of the substituted controller, without network packet loss occurring.

We implemented only the main features of BGP protocol. First, because it is a very complex protocol, which has received several updates along the years to improve it and try to correct issues found on it, although the difficult to deploy the modifications. We detected the most important characteristics and implemented on the SDN paradigm. Second, because the addition of features, i.e., the extensibility, may be easily achieved, as explained later. These features may be based on BGP implementation or may be different from it, innovating and taking more advantages of SDN benefits. On next section we present some improvements that might be done to Inter-SDN component.

Apêndice F

Final Considerations

F.1 Comments on publications

1. Bennesby, R., Fonseca, P., Mota, E., and Passito, A. (2012). An inter-as routing component for software-defined networks. In Network Operations and Management Symposium (NOMS), 2012 IEEE- This paper presents a component that enable the exchange of route information between Autonomous Systems using SDN approach to manage its domains.

2. Fonseca, P., Bennesby, R., Mota, E., and Passito, A. (2012). A replication component for resilient openflow-based networking. In Network Operations and Management Symposium (NOMS), 2012 IEEE- This paper presents the implementation and deployment of a method of passive replication to endow the SDN networks with resilience.

3. Fonseca, P., Bennesby, R., Mota, E., and Passito, A. (2013). Resilience of SDNs Based On Active and Passive Replication Mechanisms. To appear in 2013 IEEE Global Communications Conference- This paper applies a new type of replication, the active replication. The authors compare the active and passive mechanisms

applied to the SDN paradigm, analysing the behavior of each approach under certain metrics.

F.2 Conclusion

The innovation on interdomain routing may be achieved with our component thanks to the benefits provided by the SDN approach. Our experiments show that the Inter-SDN Domain Routing Component may achieve performances and scalability required for the current Internet.

The first experiment, described in Appendix E, shows that the Inter-SDN is scalable, as there is no significant difference on delay between two SDN Domains with increase of the RIB table. The second experiment of Appendix E showed that Inter-SDN is able to react to failures with little delay. We also observed that construction, deployment, maintenance, and update of the component are done in a fast and smooth way.

New features, such as multipath and source routing, may be easily added. This extensibility enables components to evolve, which is one characteristic that makes possible the interaction between Inter-SDN with other emerging applications on the SDN environment. Improvements are still necessary, but they may be easily done. For example, the search mechanism in the RIB may be improved from $\mathrm{O}(n)$ to $\mathrm{O}(\log n)$, or even a constant k, if a hash is used to manage collisions.

F.3 Future works

An interesting work involves the development of a mechanism to express routing policies on each SDN Domain and to study the effect of the application of several different policies on the relationship between different SDN Domains. The policies represent commercial interests. As competition exists, policy conflicts may happen. The design of the policy expressiveness mechanism, trying to avoid conflict issues, must follow the guidelines of current approaches, as RPSL (Alaettinoglu et al., 1997). The power of expressiveness provided by the SDN high-level abstractions is a very interesting point to be explored. Resilience will be achieved by adding a dedicated component, e.g. an extended version of Fonseca et al. (2012). In this way, we will be able to test how the performance scalability is affected on failure scenarios.

Another future work is the creation of a component able to determine a value next to optimal to the timers used to hold the announcement of route updates, as the MRAI timer, detailed before. On BGP, the MRAI timer is generally set with 30 seconds. But experiments show that the best value depend on several factors, as the topology used on domain, the traffic conditions, the used policies, among others. The SDN architecture may provide information and the level of abstraction necessary to develop efficient mechanism to discover the value closer to optimal to the timers that influence the time to domains converge to consistent states.

Referências Bibliográficas

Alaettinoglu, C., Bates, T., Gerich, E., Karrenberg, D., Meyer, D., Terpstra, M., and Villamizer, C. (1997). Routing Policy Specification Language (RPSL).

Bennesby, R., Fonseca, P., Mota, E., and Passito, A. (2012). An Inter-AS Routing Component for Software-Defined Networks. In *NOMS, IEEE*, pages 138–145.

Braga, R., Mota, E., and Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *LCN, IEEE 35th Conference on*, pages 408–415.

Bremler-Barr, A., Afek, Y., and Schwarz, S. (2003). Improved BGP convergence via ghost flushing. In *INFOCOM. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 927–937 vol.2.

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 *Proc. of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 15–28. USENIX Association.

Drutskoy, D., Keller, E., and Rexford, J. (2013). Scalable Network Virtualization in Software-Defined Networks. *Internet Computing, IEEE*, 17(2):20–27.

Elmokashfi, A., Kvalbein, A., and Dovrolis, C. (2012). BGP churn evolution: A perspective from the core. *Networking, IEEE/ACM Transactions on*, 20(2):571–584.

Feamster, N., Balakrishnan, H., Rexford, J., Shaikh, A., and van der Merwe, J. (2004). The case for separating routing from routers. In *Proc. of the ACM SIGCOMM Workshop on Future Directions in Network Architecture*, FDNA'04, pages 5–12, New York, NY, USA. ACM.

Fonseca, P., Bennesby, R., Mota, E., and Passito, A. (2012). A replication component for resilient OpenFlow-based networking. In *NOMS, IEEE*, pages 933–939.

Gao, L. (2000). Stable internet routing without global coordination. In *IEEE/ACM Transactions on Networking*, pages 681–692.

Garcia, A. (2008). *Probability, Statistics, and Random Processes for Electrical Engineering.* Prentice Hall, Toronto.

Godfrey, P., Ganichev, I., Shenker, S., and Stoica, I. (2009). Pathlet routing. *SIGCOMM Comput. Commun. Rev.*, 39(4):111–122.

Griffin, T. and Wilfong, G. (2002). On the correctness of IBGP configuration. *SIGCOMM Comput. Commun. Rev.*, 32(4):17–29.

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(3):105–110.

Halabi, S. and Mcpherson, D. (2000). *Internet Routing Architectures, second edition.* Cisco Press, Indianapolis.

Koponen, T., Shenker, S., Balakrishnan, H., Feamster, N., Ganichev, I., Ghodsi, A., Godfrey, P., McKeown, N., Parulkar, G., Raghavan, B., Rexford, J., Arianfar, S., and Kuptsov, D. (2011). Architecting for innovation. *SIGCOMM Comput. Commun. Rev.*, 41(3):24–36.

Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pages 19:1–19:6. ACM.

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.

Medhi, D. and Ramasamy, K. (2007). *Network Routing: Algorithms, Protocols, and Architectures.* Morgan Kaufmann.

Oprescu, M., Meulle, M., Uhlig, S., Pelsser, C., Maennel, O., and Owezarski, P. (2010). Rethinking iBGP routing. *SIGCOMM Comput. Commun. Rev.*, 41(4):411–412.

Passito, A. (2012). *Gerenciamento Autonomo de Redes na Internet do Futuro.* PhD thesis, Federal University of Amazonas, Brazil.

Raghavan, B., Casado, M., Koponen, T., Ratnasamy, S., Ghodsi, A., and Shenker, S. (2012). Software-defined internet architecture: decoupling architecture from infrastructure. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 43–48. ACM.

Rekhter, Y., Li, T., and Hares, S. (2006). RFC 4271: A Border Gateway Protocol 4 (BGP-4). Technical report, IETF.

Rexford, J. and Dovrolis, C. (2010). Future internet architecture: clean-slate versus evolutionary research. *Commun. ACM*, 53(9):36–40.

Vissicchio, S., Cittadini, L., Vanbever, L., and Bonaventure, O. (2012). iBGP deceptions: More sessions, fewer routes. In Greenberg, A. G. and Sohraby, K., editors, *INFOCOM*, pages 2122–2130. IEEE.

Yannuzzi, M., Masip-Bruin, X., and Bonaventure, O. (2005). Open issues in interdomain routing: a survey. *Netwrk. Mag. of Global Internetwkg.*, 19(6):49–56.

Zhang, B. (2004). Destination reachability and BGP convergence time. In *Proc. of IEEE Globecom*, pages 1383–1389.

Zhang, R. and Bartell, M. (2004). *BGP design and implementation.* Cisco Press, Indianapolis.