UNIVERSIDADE DE PASSO FUNDO

INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

CATCHMENTVIEW: FERRAMENTA DE VISUALIZAÇÃO DAS DIFERENÇAS ENTRE IPV4 E IPV6 CATCHMENT EM SERVIÇOS ANYCAST

Leonardo Costella

Passo Fundo

2018

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

CATCHMENTVIEW: FERRAMENTA DE VISUALIZAÇÃO DAS DIFERENÇAS ENTRE IPV4 E IPV6 CATCHMENT EM SERVIÇOS ANYCAST

Leonardo Costella

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Computação Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Dr. Marco Antônio Sandini Trentin
Coorientador: Prof. Dr. Ricardo de Oliveira Schmidt

Passo Fundo
2018

CIP – Catalogação na Publicação

C841c Costella, Leonardo
Catchmentview : ferramentade de visualização das diferenças entre IPV4 e IPV6 catchment em Serviços Anycast / Leonardo Costella. – 2018.
124 f. : il. color. ; 30 cm.

Orientador: Prof. Dr. Marco Antônio Sandini Trentin.
Coorientador: Prof. Dr. Ricardo de Oliveira Schmidt.
Dissertação (Mestrado em Computação Aplicada) – Universidade de Passo Fundo, 2018.

1. Software. 2. Roteadores. 3. IPV4. 4. IPV6. 5. Anycast. 6. Catchmentview. I. Trentin, Marco Antônio Sandini, orientador. II. Schmidt, Ricardo de Oliveira, coorientador. III. Título.

CDU: 004.41

Catalogação: Bibliotecária Marciéli de Oliveira - CRB 10/2113

UPF Universidade de Passo Fundo | PPGCA Programa de Pós-Graduação em Computação Aplicada
Instituto de Ciências Exatas e Geociências - ICEG

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO

LEONARDO COSTELLA

Aos seis dias do mês de dezembro do ano de dois mil e dezoito, às 10 horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo, a sessão pública de defesa do Trabalho de Conclusão de Curso "Ferramenta de visualização das diferenças entre IPv4 e IPv6 catchment em Serviços Anycast", de autoria de Leonardo Costella, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA/UPF. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Marco Antonio Sandini Trentin, Ricardo de Oliveira Schmidt, Carlos Amaral Hölbig e Roseclea Duarte Medina. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato APROVADO. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.

Prof. Dr. Marco Antonio Sandini Trentin - UPF
Presidente da Banca Examinadora
(Orientador)

Prof. Dr. Carlos Amaral Hölbig - UPF
(Avaliador Interno)

Profa. Dra. Roseclea Duarte Medina - UFSM
(Avaliadora Externa)

Prof. Dr. Ricardo de Oliveira Schmidt - UPF
(Coorientador)

Prof. Dr. Rafael Rieder
Coordenador do PPGCA

AGRADECIMENTOS

Acredito que essa seja uma das partes mais gratificantes na escrita de um trabalho. Afinal, é o momento em que posso “olhar para trás” e perceber o quanto a realização deste projeto foi enriquecedor pessoal e profissionalmente. Certamente, sem algumas pessoas esse trabalho não teria êxito, e é nesse espaço que eu gostaria de realizar os meus agradecimentos a aqueles que direta ou indiretamente contribuíram ao longo desta caminhada.

Em primeiro lugar, gostaria de agradecer a Deus pela saúde, força e coragem para superar todos os momentos difíceis em que me deparei ao longo da minha trajetória.

Aos meus familiares, em especial aos meus pais Anildo Costella e Roselaine Rovani Costella, pelo apoio incondicional durante a realização deste sonho, que não é só meu, mas nosso. Também, por todas as lições de amor, amizade, dedicação, compreensão e perdão que vocês me propiciam a cada novo dia. Sinto-me privilegiado e grato por ter pais tão especiais.

A minha namorada Adrieli, meu agradecimento por todo amor, carinho, compreensão e apoio. Obrigado por permanecer ao meu lado, me incentivar e compreender minhas corriqueiras ausências.

Ao Professor, educador, orientador e principalmente amigo Dr. Marco Antônio Sandini Trentin, meu muito obrigado de coração pelos incentivos, apoios e orientações durante toda minha vida acadêmica. Sem você meu despertar para a pesquisa jamais teria existido e este projeto não teria sido concluído.

Ao meu coorientador Dr. Ricardo de Oliveira Schmidt, pelos ensinamentos, incentivos, orientações e suporte nessa caminhada. Sem dúvidas, esse projeto só tornou-se possível com sua efetiva participação.

Aos demais professores, pela sabedoria, conhecimentos e ensinamentos perpassados durante meus estudos

Aos meus amigos e parceiros de pesquisa pelas trocas de experiências, colaboração em atividades e pelo bom convívio que tivemos, o que permitiu que essa caminhada fosse mais alegre.

E por fim, a Universidade de Passo Fundo - UPF e ao Programa de Pós Graduação em Computação Aplicada - PPGCA pelo apoio e por proporcionarem um ambiente propício para o desenvolvimento do meu projeto de mestrado.

O insucesso é apenas uma oportunidade para recomeçar com mais inteligência.

Henry Ford

CATCHMENTVIEW: FERRAMENTA DE VISUALIZAÇÃO DAS DIFERENÇAS ENTRE IPV4 E IPV6 CATCHMENT EM SERVIÇOS ANYCAST

RESUMO

Com o crescimento da adoção do IPv6, muitos serviços estão operando no formato *dual-stack*, ou seja, são configurados e aptos a receberem requisições em IPv4 e IPv6. O ideal seria que a performance em ambos os protocolos fosse a mesma. Entretanto, devido a diferenças nas políticas de roteamento, requisições em IPv4 e IPv6 para um mesmo serviço podem ser roteadas de uma maneira completamente diferente. Esse problema, acentua-se em serviços que utilizem *anycast* para compartilhar os mesmos endereços IPs entre diferentes réplicas espalhadas na Internet. Baseado no fato de que ferramentas de monitoramento de redes são importantes aliados na busca de uma melhor eficiência dos serviços disponibilizados na internet, neste trabalho desenvolveu-se uma ferramenta visual para observar e monitorar as diferenças entre o IPv4 e IPv6 *catchment* em serviços *anycast*. Essa ferramenta visa auxiliar operadores na compreensão das diferenças ocasionadas pelas políticas de roteamento, e consequentemente, colaborar na busca de melhorias na qualidade dos serviços prestados. Utilizando os servidores raiz do DNS como caso de estudo e de dados coletados da plataforma RIPE Atlas foi possível observar que a ferramenta desenvolvida neste trabalho auxiliou na constatação de discrepâncias no roteamento de usuários aos servidores raiz do DNS. Essas diferenças encontradas poderiam passar despercebidas sem o auxílio visual provido pela ferramenta.

Palavras-chave: *anycast*, *catchment*, *catchmentView*, *IPv4*, *IPv6*, *Root Servers*, *RIPE Atlas*.

CATCHMENTVIEW: A TOOL FOR VISUAL SUPPORT OF IPV4 AND IPV6 ANYCAST CATCHMENTS

ABSTRACT

Nowadays, it is common to find dual-stack services, configured with both IPv4 and IPv6 addresses. However, a service in IPv4 can be viewed and reached by users in a completely different way than its counterpart in IPv6, due to differences in routing policies for these two protocols. Although a challenge on its own, managing a dual-stack service becomes even more challenging when involving anycast: distribution of the service across the Internet by means of deployment of replicas that answer by the same IPv4 and IPv6 addresses. Based on the fact that monitoring tools have for long been strong allies in the management of networks and services in the Internet, in this work we design and implement a supporting tool for visual comparison of IPv4 and IPv6 catchments. The goal of this tool is to help operators understanding how the dynamics of their peering and routing policies impact on the catchment of their anycast service. Using the Root DNS as a case of study, with data collected by RIPE Atlas platform, we validate and demonstrate how the proposed tool supports informed decisions by the service operator.

Keywords: *anycast*, *catchment*, *catchmentView*, *IPv4*, *IPv6*, *Root Servers*, *RIPE Atlas*.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

API – Application Programming Interface.

AS – *Autonomous System.*

ASN – *Autonomous System Number.*

BGP - *Border Gateway Protocol.*

BIND – *Berkeley Internet Name Domain.*

BSD – *Berkeley Software Distribution.*

CDN - *Content Delivery Networks.*

CRAN – *Comprehensive R Archive Network.*

CSS – *Cascading Style Sheets.*

DDoS – *Distributed Denial of Service*.

DNS – *Domain Name System.*

GMT - *Greenwich Mean Time.*

HTML – *HyperText Markup Language.*

HTTP – *Hypertext Transfer Protocol.*

IANA – *Internet Assigned Numbers Authority.*

ICANN - *Internet Corporation for Assigned Names and Numbers.*

IETF – *Internet Engineering Task Force.*

IOT – *Internet of Things.*

ISP – *Internet Service Provider.*

IP – *Internet Protocol.*

IXP – *Internet Exchange Point.*

JSON – *JavaScript Object Notation.*

MX – *Mail Exchange.*

NAT – *Network address translation.*

NTP – *Network Time Protocol.*

OSPF – Open Shortest Path First.

PATRICIA – *Practical Algorithm to Retrieve Information Coded in Alphanumeric*.

PPGCA – Programa de Pós Graduação em Computação Aplicada.

RFC – *Request for Comments.*

RIP – *Routing Information Protocol*

RIPE – *Réseaux IP Européens.*

RIR – *Regional Internet Register.*

RIS – *Routing Information Service.*

RTT – *Round Trip Time.*

SSL – *Secure Socket Layer.*

TCP – *Transmission Control Protocol.*

TLD – *Top Level Domain.*

TLS – *Transport Layer Security.*

TSV – *Tab Separated Values.*

UDP – *User Datagram Protocol.*

UPF – Universidade de Passo Fundo.

URL – *Uniform Resource Locator.*

USB – *Universal Serial Bus.*

SUMÁRIO

1. INTRODUÇÃO

A rede mundial de computadores, mais conhecida como Internet (abreviação de *Interconnected Networks ou Internetwork System*) é uma rede que interliga dispositivos que se comunicam por um conjunto próprio de protocolos. Nascida para propósitos científicos, a Internet cresceu e se tornou parte fundamental na vida dos seres humanos, estando presente em grande parte das atividades que por eles são realizadas diariamente. Acessar bancos, efetuar compras, realizar encontros de forma não presencial, estudar e executar inúmeras atividades de entretenimento como *streaming* de áudio e vídeo, jogos *online* e redes sociais, são exemplos de atividades desempenhadas nos ambientes virtuais propiciados pela Internet.

Essa quase que onipresença da Internet impulsionada pelo advento dos dispositivos móveis e da Internet das Coisas (do inglês *Internet of Things* – IoT), fez com que o número de dispositivos conectados à rede mundial de computadores crescesse de forma exponencial. De acordo com a Press [1], de 2000 a 2016 o número de dispositivos conectados à Internet passou de 0.5 para 22.9 bilhões.

Todo esse aumento de dispositivos exigiu que a rede mundial de computadores passasse por algumas mudanças significativas ao longo do tempo. Dentre elas, pode-se destacar a inserção de endereçamento *anycast* em servidores, principalmente em serviços com necessidades de alta escalabilidade e o desenvolvimento de uma nova versão de endereços IP, denominada IPv6, que teve de ser projetada para suprir a falta de endereços IP disponíveis na versão 4.

Entretanto, a versão 6 do IP não pode ser considerada retro compatível com a versão 4, tornando-se necessário uma transição entre elas. A problemática que surge nessa transição é que existem diferenças entre o roteamento de pacotes IPv4 e IPv6 e essa discrepância, muitas vezes não esperada, faz com que infraestruturas tenham sua qualidade de serviço prejudicada, principalmente em serviços *anycast*.

Por ser um processo bastante dinâmico, monitorar o roteamento para grandes serviços *anycast* de forma transparente ao operador não é tarefa simples. Sem o auxílio de ferramentas visuais, é extremamente difícil e custoso saber se a qualidade dos serviços está ou não sendo prejudicada por situações decorrentes do roteamento IP.

Dessa forma, o objetivo da presente pesquisa é responder a seguinte pergunta: Como fornecer a operadores e pesquisadores informações de forma clara e precisa sobre o

estado do roteamento na Internet, em especial as diferenças entre o *IPv4 e IPv6 catchment* em serviços *anycast*?

No intuito de responder essa pergunta, o Objetivo Geral desse trabalho é desenvolver uma ferramenta de visualização capaz de auxiliar operadores de roteamento da Internet, no entendimento das diferenças entre o *IPv4 e IPv6 catchment* em serviços *anycast*.

Com vistas a atingir o objetivo principal, elaborou-se os seguintes objetivos específicos: Investigar qual a melhor maneira de apresentar as informações aos operadores; Definir quais dados são importantes a serem utilizados pela ferramenta; Coletar, filtrar e preparar os dados para serem utilizados na ferramenta de visualização; Desenvolver a ferramenta de visualização; Testar e validar a ferramenta desenvolvida com auxílios de operadores e pesquisadores.

Assim, espera-se que a ferramenta a ser descrita neste trabalho auxilie na investigação das diferenças entre o *IPv4 e IPv6 catchment* em serviços *anycast*, amparando a tomada de decisões por parte dos operadores e, consequentemente, auxiliando na busca por melhorias na qualidade dos serviços prestados.

A estrutura da dissertação está organizada em 6 capítulos. No capítulo 2 é apresentado o referencial teórico, onde são evidenciados os conceitos utilizados durante o trabalho. No capítulo 3, são abordados alguns trabalhos que se assemelham ao desenvolvido. No capítulo 4 as tecnologias utilizadas no desenvolvimento da ferramenta são retratadas. No capítulo 5, a ferramenta, suas características e um caso de uso utilizando o *Root Server* e a plataforma RIPE Atlas são exibidos. Por fim, as considerações finais são tecidas no capítulo 6.

2. REFERENCIAL TEÓRICO

Nesse capítulo serão apresentados os principais tópicos relevantes para a pesquisa. Na seção 2.1 será abordado o funcionamento do *anycast.* Na seção 2.2 será explanado a respeito do funcionamento do roteamento na Internet. Na seção 2.3 o *conceito catchment* será apresentado. Por fim, o R*oot Server*, que é o caso de estudo utilizado neste trabalho, será apresentado na seção 2.4.

2.1. IP ANYCAST

Descrito no RFC 1546 [2], *anycast* é uma técnica utilizada para compartilhar os mesmos endereços IPs entre diferentes nodos espalhados em múltiplas localizações, utilizando o protocolo BGP - *Border Gateway Protocol.* para o roteamento de clientes para esses nodos. Na figura 1, o funcionamento conceitual do *anycast* é ilustrado. O cliente simbolizado pelo círculo vermelho envia um pacote ao seu destino. Após o envio, o algoritmo de roteamento define qual cópia do destino (círculos verdes), em meio a outros serviços presentes na Internet (círculos amarelos), irá receber a requisição do cliente. Essa escolha é dada conforme a política de roteamento definida pelo BGP, sendo tipicamente utilizada a menor distância topológica entre o cliente e o destino.

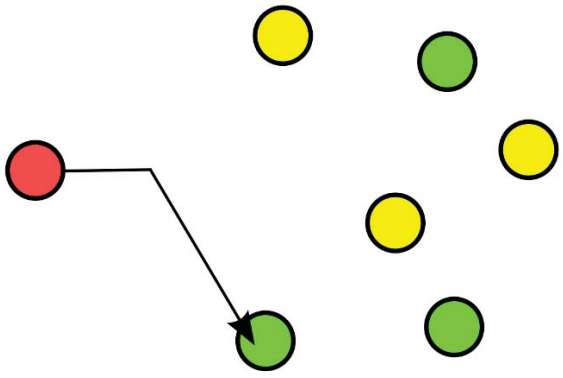

Figura 1. Ilustração do a*nycast* [3]

De acordo com Colitti *et al.*[4] as razões pelas quais tornam a utilização de *anycast* uma prática importante para a Internet nos dias atuais são:

- **Resiliência:** Caso um servidor seja afetado por ataques ou outros problemas de conexão, a distribuição do serviço prestada por esse pode ser substituída por outro servidor localizado em outro ponto do planeta. Dessa forma, o serviço prestado por aquele servidor será mantido sem que o usuário saiba por qual nodo foi atendido.
- **Performance:** Servidores localizados próximos aos clientes tendem a diminuir o tempo de requisição. Além disso, distribuir servidores ao redor do planeta é uma estratégia utilizada para dividir o tráfego de requisições, reduzindo assim a carga de trabalho por servidor.
- **Confiabilidade**: Quanto mais próximo um servidor estiver do cliente, menor será o número de saltos que uma requisição irá necessitar para percorrer o caminho entre o cliente e o servidor. Consequentemente, as chances de uma falha na transmissão da requisição também serão menores.
- **Simplicidade:** Menores cargas de requisições por servidor tornam o trabalho dos operadores menos oneroso.

A definição de *anycast* descrita no RFC 1546, é conhecida como *IP layer anycast*, e define a implementação do *anycast* na camada de rede. Entretanto, Bhattacharjee *et al.* [5] definiram um novo conceito chamado *de application-layer anycast*, que como o próprio nome sugere, opera na camada de aplicação. Este trabalho dará enfoque ao *IP layer anycast*. Dessa forma, sempre que o termo *anycast* for citado estará se referindo a esse modelo.

Uma das aplicações mais conhecidas de *anycast* é a de replicação de servidores. Nesta aplicação, a distribuição de carga entre os servidores é facilitada, visto que se o usuário ao realizar uma requisição, tem sua solicitação encaminhada a um dos *mirrors (*geralmente o mais próximo*)* responsáveis por aquele serviço. Um exemplo desse uso são os servidores de DNS - *Domain Name System.*

Outro crescente uso das técnicas de *anycast* é na localização de serviços em redes móveis. Nessa abordagem, é atribuído um endereço *anycast* a um serviço e nodos capazes de suportar os servidores deste mesmo serviço. Assim, em redes móveis de grande escalabilidade, e corriqueiras mudanças topológicas, como redes de sensores ou *ad hoc*, é possível diminuir as falhas de comunicação e o aumento da disponibilidade pode ser tão simples quanto adicionar um *host* à rede [6].

2.2. ROTEAMENTO IP

Principal função da camada de rede, o roteamento é um processo responsável por encaminhar pacotes entre a origem e o destino. Nos primeiros anos da Internet o processo de rotear pacotes era muito simples se comparado ao estado atual da mesma. O crescimento da rede mundial de computadores, bem como interesses em controlar o tráfego por questões políticas e econômicas, tornaram o roteamento IP um processo mais oneroso [7].

Peça chave no roteamento, os roteadores possuem duas funções primordiais. Uma delas é manipular cada pacote que chega, procurando uma linha de saída para ela, nas tabelas de roteamento. Esse processo recebe o nome de encaminhamento. O outro processo é responsável pelo preenchimento e atualização de tabelas de roteamento. Essas duas atividades, destacam a importância dos algoritmos de roteamento, que possuem a finalidade de descobrir o melhor caminho, entre o roteador de origem e o de destino, em um conjunto de roteadores interligados [8].

Entretanto, a Internet atual é composta por centenas de milhões de *hosts*. Armazenar informações de roteamento para cada um deles exigiria uma quantidade enorme de memória, além de que, transmitir atualizações de estados entre todos os roteadores faria com que a Internet ficasse quase que inavegável. Dessa forma, os roteadores são agrupados em A*utonomous Systems - AS* onde cada grupo de roteadores é administrado por uma mesma organização.

Esse agrupamento faz com que todos os roteadores pertencentes ao mesmo AS tenham conhecimento das informações de roteamento para cada roteador ali existente, além de executarem o mesmo algoritmo de roteamento. Para realizar contato com outros ASs, alguns roteadores presentes no AS são responsáveis por transmitir pacotes oriundos ou com destino em outros ASs. Os equipamentos com essa função são chamados de *gateway routers* [9].

Assim sendo, é possível classificar o roteamento na Internet em duas categorias: **Roteamento *intra-AS*** e **Roteamento *inter-AS.*** O primeiro determina as políticas internas de roteamento nos ASs. *RIP (Routing Information Protocol*) e *OSPF (Open Shortest Path First)* são exemplos de algoritmos *Intra-AS*. O segundo trata-se de algoritmos utilizados para determinar os caminhos ótimos entre nodos em diferentes ASs. *BGP (Border Gateway Protocol)* é basicamente, o algoritmo padrão para os roteamentos *Inter-AS.* Na figura 2 é possível visualizar um esquema que ilustra onde atuam os algoritmos de roteamento.

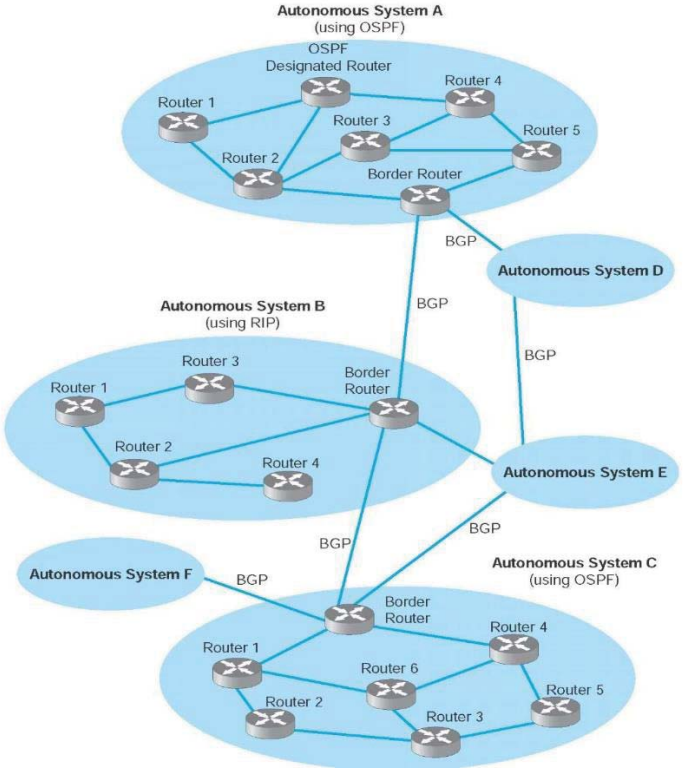

Figura 2. Esquemático do roteamento na Internet [10].

Neste esquemático, é possível visualizar como uma pequena parte da Internet pode operar. É possível verificar que, na Figura 2, cada um dos seis sistemas presentes possui um roteador de borda que o conecta aos sistemas autônomos adjacentes e troca com eles informações de rota via BGP. Neste exemplo, o sistema autônomo **A** está conectado ao sistema autônomo **B**, que por sua vez está conectado ao sistema autônomo **C**. **A** também está conectado a **C** por meio de uma rota que passa pelos sistemas **D** e **E**. Os sistemas autônomos devem compartilhar informações de rota via BGP para que os roteadores de borda em cada sistema saibam quais rotas são preferidas, cabendo ao sistema decidir qual rota irá utilizar para alcançar o destino. Também, internamente os sistemas comunicam-se utilizando os seus algoritmos internos de roteamento. Neste exemplo, **B** usa o RIP, enquanto os sistemas **A** e **C** utilizam OSPF.

2.2.1. Border Gateway Protocol – BGP

Por se tratar da proposta de uma ferramenta de monitoramento do *IPv4 e IPv6 catchment* em serviços *anycast*, esse trabalho dará destaque ao algoritmo de roteamento *inter-AS*: BGP, abstendo-se de detalhar os algoritmos *intra-AS*.

Tratado como um padrão no roteamento entre ASs, o BGP, que está na sua quarta versão[11], é o responsável por tornar visível as sub-redes existentes na Internet. Esse algoritmo oferece a cada AS meios de: obter de ASs vizinhos informações de alcançabilidade de sub-redes, propagar informações de alcançabilidade aos roteadores internos do AS e determinar as melhores rotas para as sub-redes, com base nas informações de alcançabilidade e nas políticas existentes nos ASs [9].

Na figura 3 o funcionamento do BGP é ilustrado. Os pares de roteadores que estão em diferentes ASs trocam informações por meio de uma conexão TCP (*Transmission Control Protocol)* na porta 179. Esses roteadores são denominados pares BGP e a conexão formada por eles é chamada de **eBGP**. Já a transmissão realizada entre dois roteadores do mesmo AS, ilustrada no AS 200, é denominada **iBGP**.

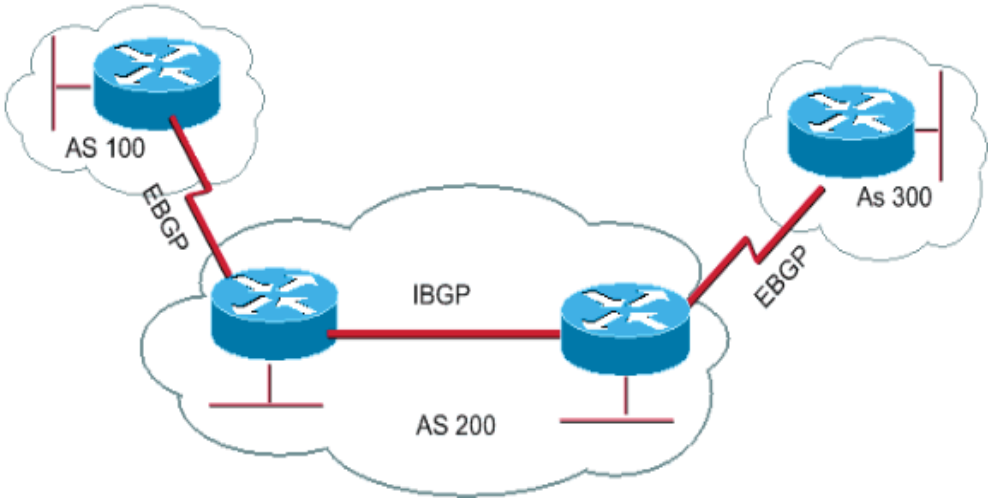

Figura 3. Ilustração do protocolo BGP [12].

Usando as sessões eBGP, os ASs trocam de forma regular informações sobre as suas tabelas de roteamento e a lista de prefixos que podem ser alcançados a partir de cada AS. Então, os roteadores de borda, por meio de sessões iBGP repassam essas informações aos roteadores internos. Os ASs são identificados no BGP por seu número exclusivo de sistema autônomo (ASN), designados pela ICANN - *Internet Corporation for Assigned Names and Numbers*[1].

Além da lista de prefixos que é anunciada por um roteador em uma sessão eBGP, vários atributos também são repassados. Os prefixos e esses atributos são chamados de **rota.** Dentre os atributos que são anunciados, dois ganham destaque: ***AS-PATH*** e ***NEXT-HOP***. O

[1] Disponível em: https://www.icann.org/

AS-PATH é o atributo utilizado para evitar *loops* no roteamento de um pacote. Esse atributo, na verdade é uma lista de números de ASs que compõem uma rota, afim de alcançar o seu destino. Quando um prefixo passa por um AS, o mesmo verifica se o seu ASN já está contido na lista. Caso esteja, ignora o anúncio, caso contrário, adiciona o seu ASN na lista. Já o NEXT-HOP é o atributo que indica qual a interface do roteador irá iniciar o AS-PATH, ou seja, qual é o primeiro salto que determinado pacote irá realizar na busca do seu destino [12].

Outra função desempenhada pelo AS-PATH é auxiliar o BGP no processo de filtragem da rota. Como mencionado anteriormente, é de responsabilidade do BPG encontrar a melhor rota entre a origem e o destino. Porém, nem sempre a melhor rota é aquela que possui o menor custo, menor distância ou menor congestionamento. Várias questões políticas estão envolvidas no roteamento e precisam ser analisadas pelo BPG. Por exemplo, é possível que um AS corporativo não esteja disposto a transportar pacotes de determinados ASs, mesmo que seu próprio AS esteja no caminho mais curto entre os dois. Por outro lado, talvez ele queira transportar pacotes para alguns ASs específicos, que tenham pago por esse serviço. Sendo assim, o BGP foi projetado para permitir a imposição de políticas de roteamento no tráfego entre os ASs [13].

2.3. IPV4 E IPV6 CATCHMENT

Durante anos o IPv4 foi o método de identificação de dispositivos exclusivo na Internet. Entretanto, percebeu-se que os 32 bits do endereçamento IPv4, resultantes em cerca de 4 bilhões de endereços, não seria suficiente para tantos dispositivos que viriam a se conectar à Internet. Dessa forma, no começo da década de 1990 a IETF (*Internet Engineering Task Force*[2]*)* concentrou esforços no desenvolvimento do sucessor do IPv4, e assim surgiu o IPv6.

Em fevereiro de 2011, a IANA (*Internet Assigned Numbers Authority)*[3] alocou o último conjunto de endereços IPv4. Porém, nessa época a implementação do IPv6 ainda era pequena, onde apenas 8.28% dos *ASs* anunciavam prefixos nessa versão [14]. Mesmo com o esgotamento de endereços IPv4 e a ainda baixa adesão ao IPv6, o número de *hosts* na Internet continuou crescendo graças a mecanismos como o NAT [15].

O *Network Address Translation* - *NAT* é um protocolo utilizado para anunciar somente um endereço para toda uma rede ao mundo exterior. Para isso, o NAT realiza um mapeamento baseado no IP interno e na porta local do computador. Com esses dois dados, o

[2] Disponível em: https://www.ietf.org/.
[3] Disponível em https://www.iana.org/.

NAT gera um número de 16 *bits*, para identificar determinado computador na rede interna. Dessa forma, toda vez que um pacote for enviado para fora desta rede, levará o IP global do *router* e na porta de origem o número que foi gerado pelo NAT. Assim, quando o *NAT* receber a resposta, saberá, realizando a operação inversa, para qual dispositivo irá a destinar. Ou seja, procurando em sua tabela o IP interno correspondente aos *bits* do campo da porta, o *NAT* direciona o pacote ao IP correspondente.

O IPv6 possui endereços de 128 *bits* e surgiu para resolver o problema de escassez de endereços IPv4, mas também para proporcionar uma maior segurança, mobilidade, suporte e tornar a configuração da Internet mais simples. Até a realização deste estudo, 24.79% dos ASs já estavam operando em IPv6 [14].

Como categorizou Nikkhah e Guerin [16] o IPv6 está em uma fase onde já é possível confiar que a performance de roteamento atual desse protocolo é semelhante ao IPv4. Entretanto, Dhamdhere *et al* [17] provou em suas análises que a diferença de performance do serviço em IPv6 é pior quando o caminho entre o IPv4 e o IPv6 é diferente. Essa diferença acentua-se quando se trata de serviços DNS, visto que muitos protocolos necessitam do DNS para funcionar de forma correta, como por exemplo e-mail, websites e muitos outros.

Uma das maneiras de realizar comparativos de roteamento em redes de computadores é gerando, de forma gráfica, representações da região topológica em que pacotes oriundos de diferentes *hosts* são roteados até o seu destino, em especial em serviços *anycast*, onde requisições de diferentes locais para um mesmo destino podem ser recebidas em diferentes pontos. Nesse sentido, vários trabalhos, inclusive este, utilizam de forma analógica o termo *catchment* para descrever a região topológica de uma rede na qual os pacotes de diferentes *hosts* são direcionados a um serviço em comum que utiliza *anycast.*

O termo *catchment* é oriundo de estudos geográficos, em especial para representações de bacias hidrográficas. Nesse contexto, o *catchment* é a representação da área de terra por onde a água da chuva flui sobre a paisagem até encontrar seu destino, que eventualmente é um rio. Um exemplo de *catchment* hidrográfico é ilustrado na Figura 4.

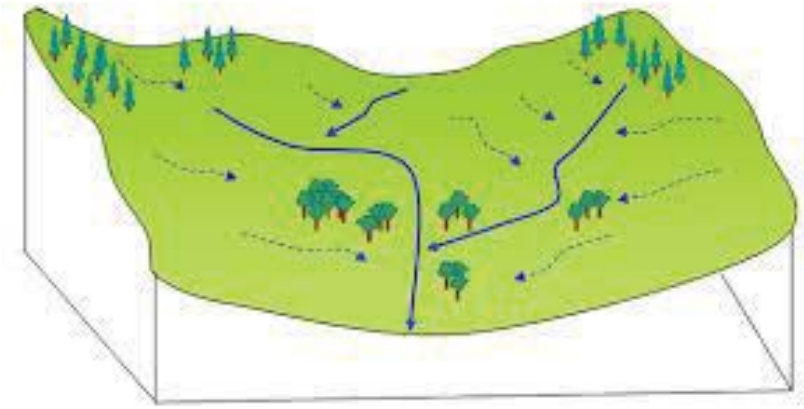

Figura 4. Exemplo de *catchment* hidrográfico [18].

O *catchment* na Internet, é tipicamente definido pelas regras presentes no protocolo BGP. Se a configuração presente no BGP não for cuidadosamente planejada, o *catchment* de determinados serviços pode não ser o pretendido. Nessas condições, provavelmente resultarão em performances não otimizadas no roteamento de pacotes, causando problemas à qualidade dos serviços prestados.

2.4. ROOT SERVERS

Os *Root Servers*, caso de estudo deste trabalho, são servidores distribuídos globalmente e que são vitais para o funcionamento do DNS. Protocolo da camada de aplicação, o *DNS* [19], diferentemente dos demais protocolos dessa camada, não é uma aplicação a qual o usuário interage diretamente. Responsável por traduzir os nomes de hospedeiros *(hostname)* para endereços IP e vice-versa, o DNS funciona como um banco de dados distribuído executado em uma hierarquia de servidores, a qual é representada na Figura 5.

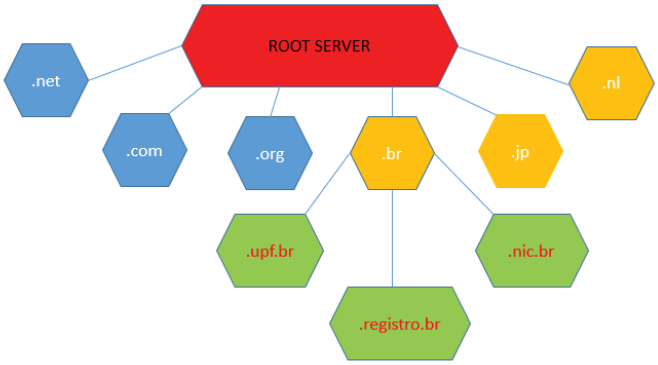

Figura 5. Exemplo de hierarquia DNS.

Em geral, os servidores DNS são computadores que executam o software *BIND (Berkeley Internet Name Domain)*, que foi criado por quatro estudantes de graduação da Universidade de Berkeley e distribuído pela primeira vez com o sistema operacional 4.3 BSD. Atualmente o mesmo é suportado e mantido pelo *Internet Systems Consortium*, estando em sua nona versão [20].

Além da tradução de *hostnames* para endereços IP, o DNS possui outras importantes funções [9]:

- **Apelidos de hospedeiro:** Alguns *hosts* podem possuir nomes complicados. Nesse caso, é possível atribuir alguns apelidos a eles. Os apelidos são denominados *alias* e o nome original do *host* é chamado de *canonical name*.
- **Apelidos para servidores de correio:** Assim como os *hosts*, é adequado que endereços de e-mail sejam fáceis de lembrar. O DNS pode ser chamado por uma aplicação para obter o *canonical name* a partir de um apelido fornecido. Além disso, o registro MX (*Mail Exchange)* também é responsável por permitir, por exemplo, que o servidor de correio e o servidor Web de uma empresa tenham nomes idênticos.
- **Distribuição de carga:** Sites com volumosos acessos podem possuir cópias replicadas em diferentes servidores com diferentes endereços IP. O DNS ao receber constantes consultas, realiza uma espécie de rodízio, a fim de distribuir o tráfego entre os servidores replicados. Esse rodízio também pode ser usado com servidores de e-mails.

2.4.1. Banco de dados distribuído

Devido ao tamanho da Internet e ao crescente aumento no número de *hosts* conectados, armazenar todos os mapeamentos em um único servidor DNS tornar-se-ia inviável. Alguns problemas de um arranjo centralizado seriam a baixa disponibilidade, o grande volume de tráfego gerado a um mesmo servidor, a distância entre alguns *hosts* e o servidor DNS e a onerosidade na manutenção deste único servidor.

Com o intuito de não gerar esses problemas, o DNS trabalha com um grande número de servidores organizados hierarquicamente e distribuídos ao redor do planeta, partilhando o mapeamento entre todos os servidores DNS existentes. Esse sistema hierárquico funciona com três classes de servidores DNS: Root, *top-level domain - TLD* e servidores DNS autoritativos.

Os ***Root servers*** são formados por treze conjuntos de servidores distribuídos globalmente (denominados de A a M) e operados por doze organizações independentes. Esses servidores são responsáveis por responder as requisições dos servidores TLDs. Até o segundo semestre de 2017, haviam 762 servidores DNS espalhados por todo o globo. Destes, 683 (89,6%) operando em IPv6 [21]. Na tabela 1, são apresentados os endereços IPs dos *Root Servers* e a data em que o IPv6 entrou em funcionamento nos mesmos.

Tabela 1. Informações sobre os *Root Servers*

Root Server	IPv4	IPv6	Data IPv6[4]
A	198.41.0.4	2001:503:ba3e::2:30	02/2008
B	192.228.79.201	2001:500:200::b	08/2015
C	192.33.4.12	2001:500:2::C	03/2014
D	199.7.91.13	2001:500:2D::D	03/2014
E	192.203.230.10	2001:500:a8::e	08/2016
F	192.5.5.241	2001:500:2f::f	02/2008
G	192.112.36.4	2001:500:12::d0d	10/2016
H[5]	198.97.190.53	2001:500:1::53	02/2008
I	192.36.148.17	2001:7fe::53	06/2010
J	192.58.128.30	2001:503:c27::2:30	02/2008
K	193.0.14.129	2001:7fd::1	02/2008
L	199.7.83.42	2001:500:9f::42	12/2008
M	202.12.27.33	2001:dc3::35	02/2008

Os **TLDs** são os servidores DNS responsáveis pelos domínios de alto nível como, por exemplo, .com, .edu, .org, .net, .gov e pelos domínios de alto nível dos países como, por exemplo, .br, .pt, .uk e .jp. A lista de completa de todos os domínios de alto nível encontra-se disponível em [22]. Cada TLD conhece os endereços dos servidores DNS autoritativos, que pertencem ao seu domínio, ou o endereço de algum servidor DNS intermediário que conhece um servidor autoritativo.

Toda organização que possui hospedeiros disponíveis a serem acessados na Internet, deve tornar público os seus registros DNS que associam aquele domínio a seu endereço IP. O **DNS autoritativo** é o servidor que está autorizado a responder por um domínio. Essa

[4] http://www.root-servers.org/news.html
[5] Não operava em modo *anycast* até a realização deste trabalho.

autoridade é dada pelos registros de domínios oficiais, de uma determinada região do mundo, autorizados pelos seus correspondentes *RIRs - Regional Internet Register.*

Cada *Internet Service Provider – ISP* ou organizações, tais como o de uma universidade ou empresa, possui um servidor DNS local. Toda vez que um *host* faz uma consulta DNS, ela é enviada ao **servidor DNS local** que age como um *proxy* e a retransmite para a hierarquia do servidor DNS.

Os servidores DNS são um bom exemplo de uso de *anycast*. Além da distribuição de carga, a busca de resiliência em casos de ataques ou falhas impulsiona o uso desta técnica. Por exemplo, através de cópias de serviço espalhadas em diferentes localizações do globo terrestre, ataques de negação de serviço (*DDoS - Distributed Denial of Service)* podem ser contornados, desviando o tráfego que antes iria aos servidores atacados para algum servidor não afetado.

Essa busca por resiliência ganhou força em 21 de Outubro de 2002 [23]. Nessa data, um *DDoS* deixou 9 dos 13 *Root Servers* inacessíveis. Ainda, em Fevereiro de 2007 um novo ataque aos *Root Servers* com menores consequências foi relatado. Na ocasião, 6 *Root Servers* foram alvos, entretanto apenas dois, que ainda não haviam implantado *anycast*, foram visivelmente afetados [24]. Hoje, apenas o R*oot Server H* não utiliza *anycast* e mais de 72% dos *TLD servers* já trabalham com replicações de seus serviços [25].

2.4.2. Tradução de *hostnames*

Um *host* na Internet pode ser identificado de duas maneiras: por seu *hostname* ou pelo endereço IP correspondente ao equipamento que hospeda determinado serviço. A camada de rede utiliza os endereços IPs para a localização desses equipamentos, todavia, nomes de domínio são mais apreciados pelos usuários, por possuírem uma melhor mnemónica[6]. É nesse sentido que trabalha a principal função do DNS, que é a de traduzir nomes em endereços IP.

Sempre que um *host* necessitar acessar um endereço qualquer (“www.ripe.net”, por exemplo), os seguintes passos são necessários para encontrar o endereço IP que corresponde ao *hostname* requerido:

1. O *resolver* (cliente DNS) envia uma requisição para os servidores de nomes local.

[6] A mnemónica é o processo de estabelecer uma associação para relembrar alguma coisa.

2. O servidor local de posse da URL, envia uma consulta para o R*oot Server*, que analisa a URL e percebe o sufixo “net”.
3. O *Root Server* responde com uma lista de endereços IP para o DNS local, contendo servidores TLDs que são responsáveis pelo sufixo “net”.
4. De posse do endereço dos TLDs, o DNS local faz uma requisição a algum desses servidores.
5. Ao reconhecer “ripe.net”, o servidor TLD retorna ao DNS local o endereço IP do servidor que detém autoridade para RIPE.
6. Em seguida o DNS local envia uma consulta ao servidor com autoridade.
7. O servidor autoritativo responde com o endereço IP do hospedeiro “www.ripe.net”.
8. Por fim, o DNS local envia ao *host* o endereço IP do hospedeiro www.ripe.net.

Com esse exemplo, é possível verificar a existência de consultas **iterativas** e **recursivas**. A consulta recursiva acontece quando o *host* requisita que o DNS local mapeie o endereço IP em seu nome. Já as interações entre o DNS local com o DNS raiz, o DNS autoritativo e o TLD são consultas iterativas, uma vez que a resposta é retornada de forma direta ao DNS local.

Outro ponto a se destacar nas consultas DNS é a existência da memória *cache*. A *cache* é um recurso massivamente usada em servidores DNS. Tomando como base o exemplo acima, onde foi ignorada a existência dessa ferramenta, oito trocas de mensagens são necessárias até que o endereço IP correspondente a URL requerida seja encontrada. Apesar dessa troca de mensagens ocorrer de forma relativamente rápida, o mesmo processo pode ser agilizado fazendo o uso de *caches* de DNS.

Sempre que um servidor DNS recebe uma resposta, o mesmo pode fazer um c*ache* das informações contidas na resposta em sua memória local. Se uma requisição de tradução de nomes chegar até um servidor e as informações estiverem salvas na memória *cache*, o mesmo poderá imediatamente responder o requerente, sem realizar uma nova consulta. Isso ocorre mesmo que o servidor não tenha autoridade para aquele nome.

3. TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos que se assemelham com o trabalho proposto. Nas sessões 3.1 e 3.2 duas ferramentas de visualização de resultado de traceroute, intituladas RADIAN e TRACEMON, respectivamente, serão apresentadas. E por fim, na sessão 3.3 será retratado um estudo sobre as diferenças entre o *catchment* IPv4 e IPv6 em serviços *anycast* que serviu de motivação para o desenvolvimento dessa ferramenta.

3.1. RADIAN

Em 2016, o *Graph Drawing and Network Visualization group* da Universidade de Roma Tre na Itália desenvolveu uma ferramenta para análise virtual de *traceroute* utilizando medições que podem ser realizadas em tempo real. A ferramenta intitulada *Radian*[7], está disponível de forma online, em uma versão demo que permite utilizar até no máximo 5 *probes* e possibilita que o usuário explore os dados interagindo e analise a evolução do roteamento por meio de animações [26].

Na figura 6, é possível observar a ferramenta *Radian* em uso. No cenário, as *probes* executam periodicamente o *traceroute* em direção a um alvo fixo. O software mescla os caminhos resultantes em um gráfico, que é visualizado com as *probes* na periferia do desenho (representadas como círculos azuis) e o alvo no centro (representado como um círculo vermelho).

Os caminhos do *traceroute* são representados como caminhos coloridos dos *probes* para o destino. Enquanto caminhos tracejados representam um *traceroute* que não mudou com o tempo, linhas sólidas envolveram algumas dinâmicas no roteamento. O usuário pode rever a dinâmica clicando no mouse na parte inferior da página. Um evento no roteamento é mostrado com uma animação de um caminho desde sua posição inicial até a final.

Os endereços IP são agrupados em AS, representados como uma caixa azul quando contem *probes* e por caixas amarelas quando são apenas AS para trânsito. O usuário pode clicar duas vezes em um AS para expandi-lo ou reduzi-lo, e assim escolher a quantidade de detalhes

[7] Ferramenta disponível em: http://www.dia.uniroma3.it/~compunet/projects/radian/

a ser exibida. Além disso, clicar com o botão *shift* em uma *probe*, o *roundtrip – RTT,* daquele caminho é exibido.

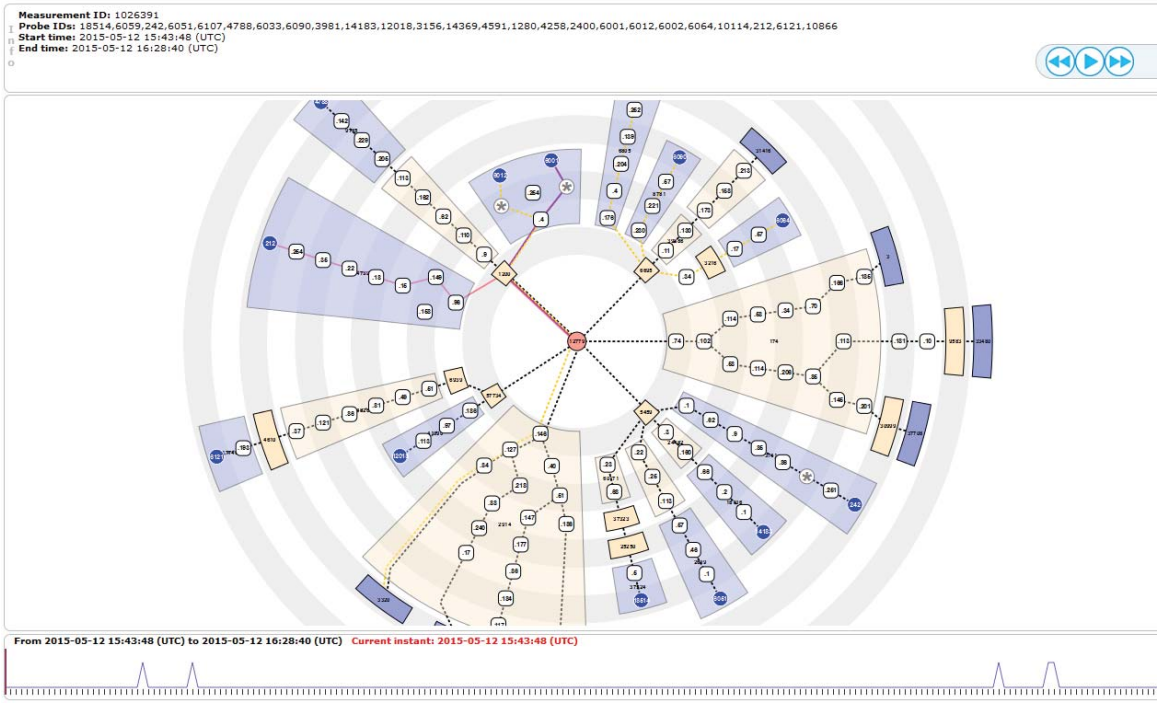

Figura 6. *Radian interface* [27].

3.2. TRACEMON

TraceMON [28] é uma ferramenta utilizada no monitoramento e investigação dos resultados de medições *traceroute*. Com ela é possível visualizar algumas características e componentes envolvidos nos roteamentos utilizados como entrada, como por exemplo: contato dos responsáveis pelos ASNs, latência, *whois*, geolocalização de IPs, entre outros.

Utilizando as medições do projeto RIPE Atlas ou repositórios de terceiros para inferir a topologia de rede e características dos vários componentes de rede envolvidos, com o *TraceMON* é possível depurar instantaneamente determinadas situações envolvidas no roteamento.

Conforme pode ser observado na Figura 7, vários resultados do *traceroute* são representados em um gráfico onde as *probes* são representadas pelos nós verdes, situados na parte superior do gráfico. Os nós que representam os *IXPs - Internet Exchange Point* são exibidos na cor azul, em cinza os IPs não encontrados e em amarelo os demais IPs que envolvem o *traceroute.*

Todos os nós são rotulados e ao serem clicados, informações detalhadas sobre cada nó são exibidas, incluindo eventos do BGP coletados pelo serviço do RIS (*Routing Information Service)* ou endereços de e-mail do contato técnico. Se uma fonte não puder executar o *traceroute*, um pequeno símbolo de erro vermelho aparecerá. Além disso, clicando em um caminho é possível recuperar a saída de *traceroute* textual usada para a representação do gráfico.

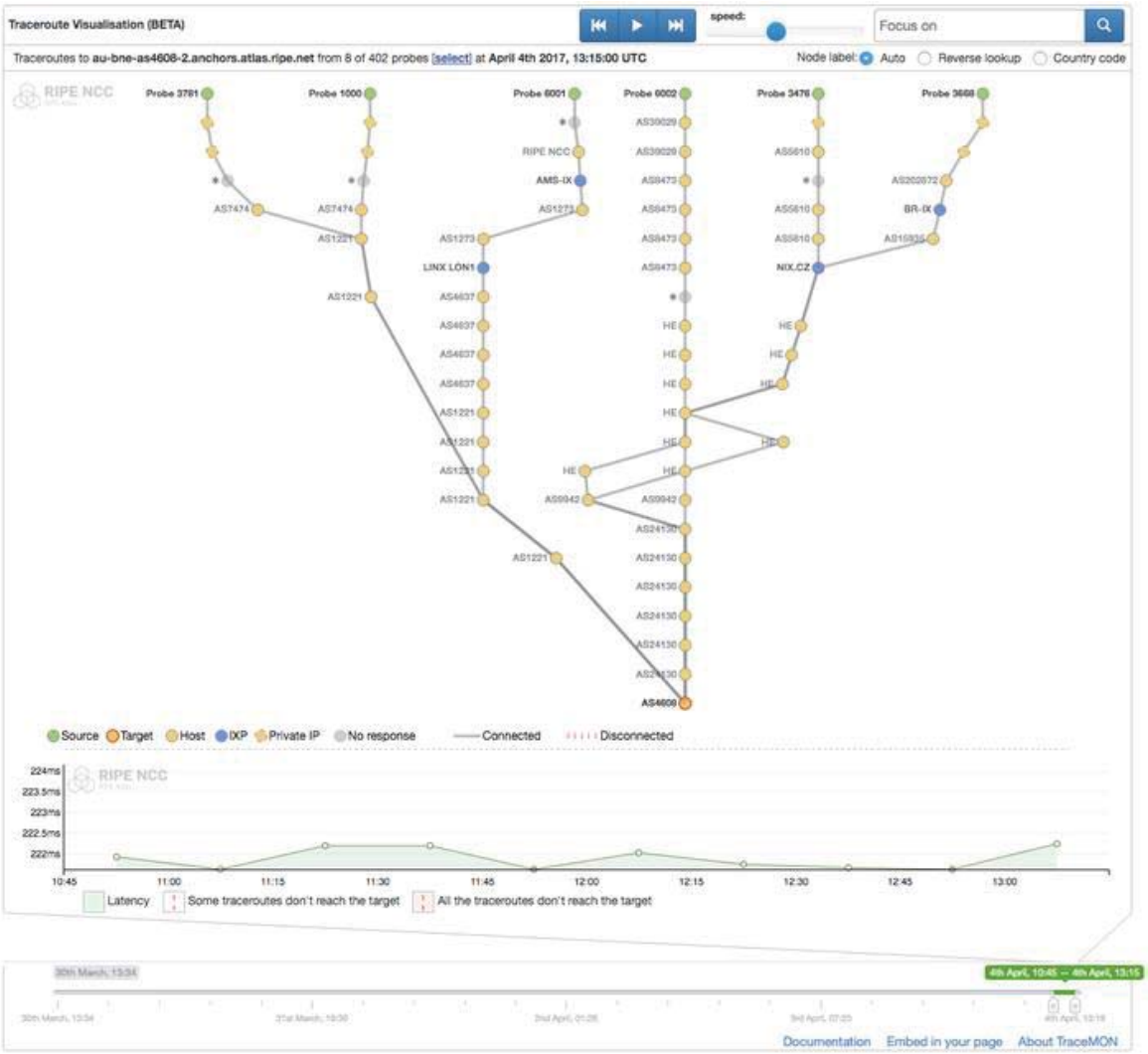

Figura 7. TraceMON interface [28].

O *TraceMON* seleciona um conjunto de *probes* por padrão com base em uma análise preliminar dos *metadados*. Alternativamente, um conjunto diferente de *probes* pode ser selecionado manualmente, além de ser possível dar ênfase a determinado *traceroute.*

3.3. IPV4 VS IPV6 ANYCAST CATCHMENT: A ROOT DNS STUDY

Wicaksana [29] analisou as diferenças entre os *catchments IPv4 e IPv6* de 9 R*oot Servers* que operavam em modo *anycast*, com dados oriundos do BGP RIPE RIS de Fevereiro de 2008 a Junho de 2016. Esse estudo chegou à conclusão de que existem evidentes diferenças no roteamento de requisições realizadas para serviços IPv4 e IPv6 *nos Root Servers anycasted* que operam em modo *Dual-Stack.*

Na Figura 8 é apresentado um exemplo das análises presentes no trabalho em questão. Nesta, o objetivo é observar de forma percentual as diferenças entre os caminhos percorridos em IPv4 e IPv6. Na parte inferior do gráfico são apresentadas as datas em que os dados foram gerados e na parte superior o número de pontos de coleta utilizados naquele período. As barras na cor azul representam a porcentagem de consultas que obtiveram caminhos diferentes em IPv4 e IPv6. Em verde as que obtiveram caminhos idênticos. Em vermelho é representada as consultas em que o caminho percorrido pelo IPv4 foi menor. E em verde claro as consultas em que o caminho percorrido pelo IPv6 foi menor.

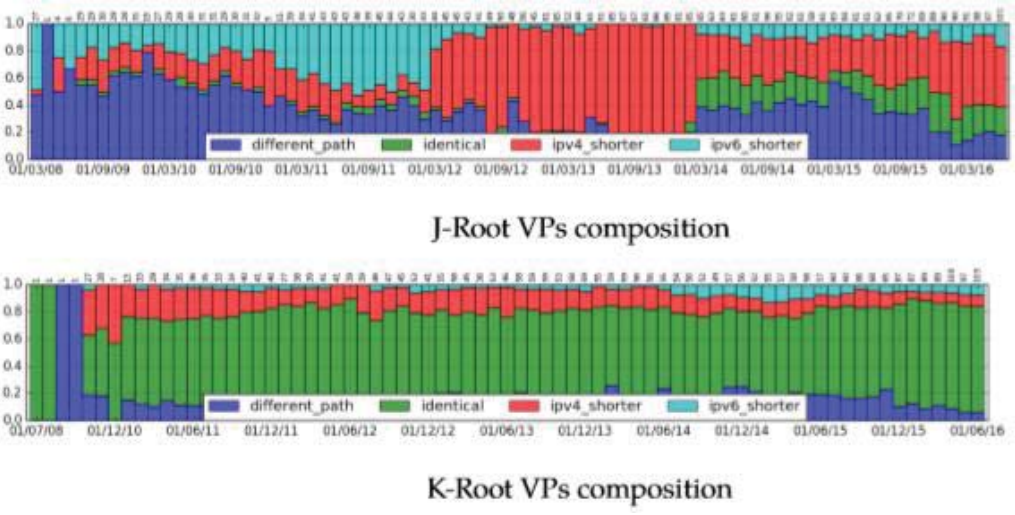

Figura 8. Exemplo de análise IPv4 vs IPv6 *catchment* [29]

Além disso, para uma melhor compreensão dos caminhos percorridos, gráficos interativos foram desenvolvidos. Na Figura 9 é apresentado um exemplo desses gráficos que, de acordo com o autor, possuem o objetivo ilustrar de forma interativa as diferenças entre o *IPv4 e IPv6 catchment* dos serviços *anycast* estudados e expor maiores informações referentes aos componentes envolvidos nos *traceroute* em questão.

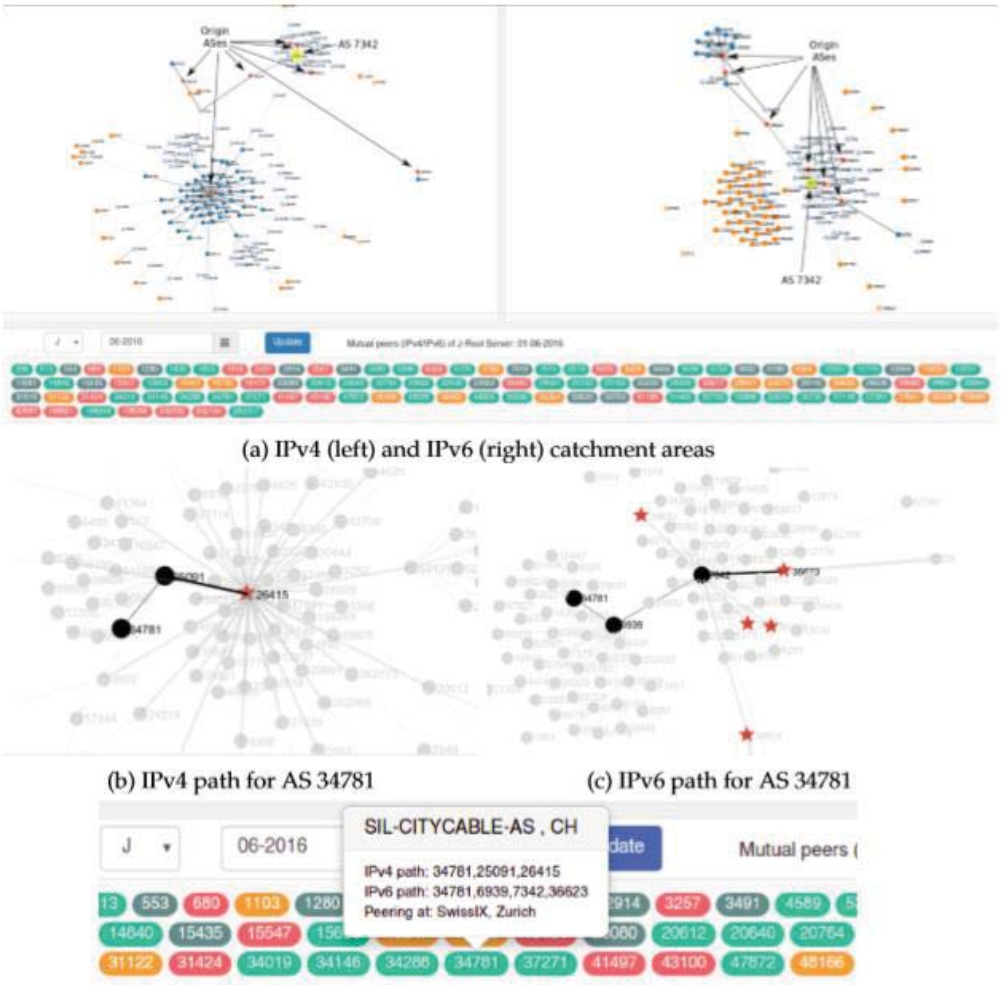

Figura 9. *Catchment* interativo [29].

Os resultados obtidos nesse trabalho reforçaram a necessidade de se tomar conhecimento das diferenças presentes no IPv4 e IPv6, em especial no que diz respeito ao roteamento para serviços *anycast*. Essas análises serviram de apoio ao desenvolvimento da ferramenta de visualização, a ser descrita posteriormente no capítulo 4.

4. TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentadas as principais tecnologias utilizada na construção da ferramenta, que será descrita no capítulo 5. Na decisão das tecnologias a serem usadas, optou-se por escolhas que permitissem uma simplicidade no desenvolvimento, além da possibilidade de se obter escalabilidade e ampliação das funcionalidades presentes. As tecnologias listadas a seguir supriram as necessidades e foram as principais empregadas no desenvolvimento da ferramenta.

4.1. R

R é uma linguagem de programação altamente extensível e voltada especialmente à computação estatística, manipulação de dados e exibição gráfica. Disponível como um software livre, sobre os termos da *Free Software Foundation's GNU General Public License*, a linguagem R foi desenvolvida na Universidade de Auckland, Nova Zelândia, por Ross Ihaka e Robert Gentleman [30]. Similar a linguagem S, o R oferece uma grande variedade de estatísticas (modelagem linear e não-linear, testes estatísticos clássicos, análise de séries temporais, classificação, agrupamento, etc.) e técnicas gráficas extensíveis, que fornecem uma rota de código aberto para que haja participações entre programadores.

Um dos pontos fortes do R é seu ambiente de desenvolvimento, o qual possui um grande número de algoritmos que que facilitam a manipulação de dados, os cálculos e a visualização gráfica. Dessa forma, o ambiente de desenvolvimento R pode ser estendido facilmente por meio de pacotes. Existem mais de 13 mil pacotes disponíveis na família *CRAN - Comprehensive R Archive Network*[8], além de vários outros disponibilizados em demais projetos ou por desenvolvedores em seus repositórios. Também, para tarefas computacionalmente intensivas, códigos em *C, C ++* e *Fortran* podem ser vinculados e chamados em tempo de execução no ambiente R, podendo inclusive manipular objetos R diretamente [31].

[8] O CRAN é uma rede de servidores *ftp* e web em todo o mundo que armazena versões idênticas e atualizadas de código e documentação para R – Documentação disponível em: https://cran.r-project.org/

Nesse projeto, o ambiente e linguagem R foram utilizados no desenvolvimento da ferramenta. Pelas características já acima elencadas, essa tecnologia foi a escolhida a ser utilizada desde a manipulação dos dados até a elaboração do ambiente de visualização gráfica.

4.2. PACOTE SHINY DASHBOARD

Shiny é um pacote *R open source* responsável por usufruir do potencial computacional do R para construir aplicações web interativas. Com esse pacote é possível utilizar uma variedade de *widgets* disponíveis, além de mesclar códigos em *HTML*, *CSS*, *JavaScript* e *JQuery* para estender as possibilidades da aplicação web [32].

Na ferramenta de monitoramento aqui descrita, utilizou-se o pacote *ShinyDashboard* para o desenvolvimento da *user interface*. O pacote *ShinyDashboard* [33] é uma extensão do pacote Shiny, o qual baseia-se no tema *AdminLTE Bootstrap* para desenvolver aplicações mais completas, utilizando componentes *BootstrapUI* [9].

4.3. FORCE-DIRECTED GRAPH

Force-directed graph são classes de algoritmos utilizados para o desenvolvimento de gráficos de uma maneira esteticamente agradável. Para tal, o algoritmo posiciona os nós, seja em um espaço bidimensional ou tridimensional, de modo que todas as arestas sejam de comprimento mais ou menos igual e haja o menor número possível de arestas de cruzamento, atribuindo forças entre o conjunto de arestas e o conjunto de nós com base em suas posições relativas [34].

Métodos de *Force-directed graph* surgiram em 1963, quando Tutte mostrou que *gráficos poliédricos*[10] podem ser desenhados no plano com todas as faces convexas, fixando os vértices da face externa do gráfico em posição convexa, colocando uma força atrativa, semelhante a uma mola, em cada borda e deixando o sistema estabelecer um equilíbrio [35]. Essa teoria serviu como base para o trabalho de Eades, que em 1984 usou uma combinação de forças de atração em vértices adjacentes e forças repulsivas em todos os vértices na confecção de seus grafos [36].

---

[9] Bootstrap é um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web

[10] Um grafo poliédrico é o grafo não direcionado formado a partir dos vértices e arestas de um poliedro convexo.

O algoritmo começa por anexar um ponto elétrico fictício para os nós. Da mesma maneira, presume-se que cada ramo da rede contém uma mola fictícia, assim, se tem um sistema eletromecânico, no qual as cargas elétricas tentam repelir cada nó para longe de todos os outros nós, ao passo que molas mecânicas tentam trazer os nós que estão conectados com filiais mais próximas. Eventualmente, o sistema se instala em um estado estável de baixa energia que corresponde para uma representação esteticamente agradável do gráfico [37]. Além disso, outras forças, como por exemplo uma força análoga à gravidade, podem ser utilizadas para puxar vértices em direção a um ponto fixo do espaço de desenho, no intuito de unir diferentes componentes conectados de um gráfico desconectado, que de outra forma tenderiam a se separar um do outro por causa das forças repulsivas, e também, para auxiliar a desenhar nós com maior centralidade [38].

Na Figura 10 é possível visualizar a comparação entre o *layout* sem os conceitos do *force-directed graph* e após a utilização dos mesmos. Esse experimento utilizou 42 nodos que representam o sistema de ônibus da rede Regional Ocidental da Índia [37]. De acordo com o próprio autor, pode-se observar que o algoritmo foi capaz de melhorar o *layout* final, pois contém um menor número de cruzamentos se comparado ao *layout inicial*.

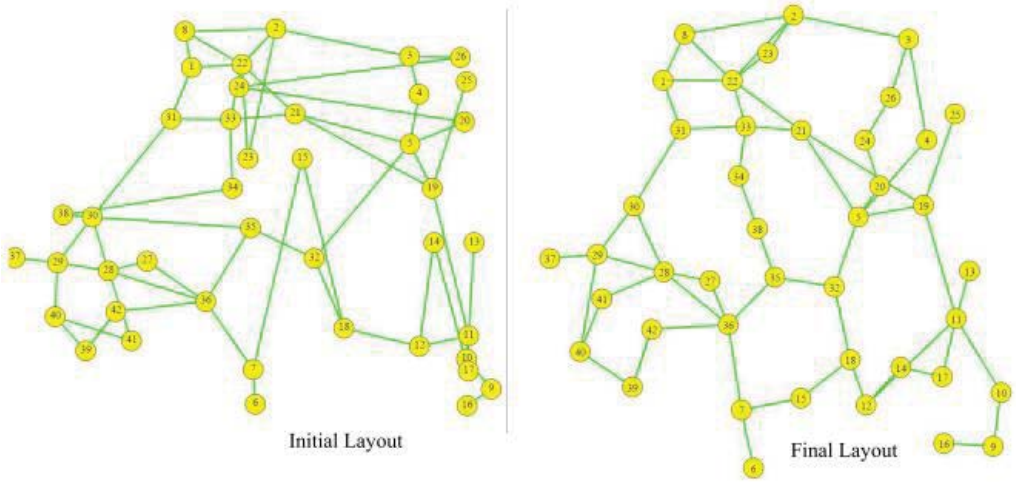

Figura 10. *Layout* de um gráfico antes e depois dos conceitos do *force-directed Graph* [37].

Na ferramenta de visualização a ser descrita no capítulo 5, se faz uso de um *force-directed graph* na montagem dos gráficos que ilustram o *catchment.* A escolha por esse tipo de gráfico se deu pela forma esteticamente agradável que esse tipo de grafo se apresenta, possibilitando uma fácil compreensão da situação a ser exibida, além de uma boa aceitação à

inserção de interações nos nodos e vértices que compõem o gráfico, devido a menor probabilidade de ocorrerem cruzamentos entre os vértices.

Utilizando o pacote *NetworkD3*[39] , adaptou-se para as necessidades da ferramenta a versão do algoritmo *forceNetwork* presente na mesma, o qual é o responsável por gerar o *force-directed graph*. O pacote *Network D3* é um pacote disponível pela CRAN que visa o desenvolvimento de interativos gráficos para representações de redes, utilizando a linguagem R.

4.4. RIPESTAT API

Desenvolvida pela RIPE NCC[11], a *RIPEstat* é uma *API (Application Programming Interface*) que provê informações sobre endereços de IP, *Autonomous System Numbers (ASNs)* e dados relacionadas a nomes de *host* e países.

Na ferramenta desenvolvida neste trabalho, a ser apresentada no capítulo 5, a plataforma *RIPEstat* é utilizada para encontrar a localização do IP correspondente a cópia do serviço *anycast* que respondeu determinada requisição, no momento em que a requisição foi realizada. De posse da localização, utilizando a função de *Haversine*, a ser descrita na sessão 4.5, é possível encontrar qual é a cópia do serviço *anycast* que está mais próxima da localização retornada pela *RIPEStat API*.

As informações de geolocalização da *RIPEstat* são oriundas dos dados do *GeoLite*[12] criado e mantido pela *MaxMind*. De acordo com a própria *MaxMind* [40], os seus bancos de dados não são 100% precisos. Em virtude disso, por utilizar o método de *geolocalização* para definir qual cópia do serviço *anycast* foi responsável por receber determinada requisição, o trabalho aqui apresentado irá tratar essa cópia como a provável cópia, respeitando assim, as limitações existentes neste método.

4.5. FÓRMULA DE HAVERSINE

A fórmula *Haversine* é uma fórmula amplamente utilizada para determinar a menor distância entre dois pontos em uma esfera, dadas as suas longitudes e latitudes. Importante na navegação, é um caso especial de uma fórmula mais geral em trigonometria esférica, a lei de Haversines, que relaciona os lados e ângulos dos triângulos esféricos. Os primeiros registros da

[11] Réseaux IP Européens Network Coordination Centre. Informações em: https://www.ripe.net/
[12] Banco de dados Geolite, disponível em: https://dev.maxmind.com/geoip/legacy/geolite/

fórmula de Haversine com esse nome são de James Inman em seu livro *"Navigation and Nautical Astronomy for the Use of British Seamen"*, escrito em 1849 [41]. Entretanto, existem outros relatos do uso da fórmula, nos trabalhos de James Andrew em 1805 [42] e José de Mendoza y Ríos em 1801 [43].

A fórmula de Haversine é definida como [44]:

$$haversine(\theta) \equiv \sin^2\left(\frac{\theta}{2}\right) \tag{1}$$

Considerando dois pontos de uma esfera de raio *R*, com latitudes e longitudes (ϕ_1, λ_1), (ϕ_2, λ_2), respectivamente, a distância *d* é:

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1)}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1)}{2}\right)}\right) \tag{2}$$

A função *distHaversine do* pacote *geosphere* [45] é utilizada para obtenção da menor distância entre dois pontos, seguindo o método de *Haversine*.

4.6. IP TO ASN

IPtoASN é um banco de dados utilizado para encontrar a qual *Autonomous System* pertence determinado IP. Desenvolvido por e mantido por Frank Denis, o banco de dados *IPtoASN* é atualizado de hora em hora e disponibilizado para *download*[13] no formato *TSV (Tab Separated Values)*, contendo o *IP range* (IP inicial e IP final*), o ASN* que aquele bloco de *IPs* pertence, o código do país daquele *AS e* uma descrição do *Autonomous System*. Essas informações são adicionadas a uma estrutura de dados chamada *Radix Tree*, a qual é utilizada para se obter menores tempos de busca pelos *ASN*.

4.7. RADIX TREE

A estrutura de dados *radix tree*, também conhecida por *radix trie, compact prefix tree ou PATRICIA trie* é uma estrutura de dados que representa uma *trie* otimizada no que diz respeito ao armazenamento, visto que na *radix trie* cada nó que é o único filho é mesclado com seu pai, o que reduz o custo de armazenamento em relação a outras árvores, como a *Trie tree*

[13] IP to ASN Data Base. Disponível em: https://iptoasn.com/

por exemplo, onde para representar um nodo existente na árvore, podem ser necessários uma maior quantidade de nodos internos [46]. Na Figura 11 é exemplificada a diferença na quantidade e disposição dos nodos entre uma *Trie Tree* e a *Radix Tree* com 5 *strings* adicionadas.

Concebida por Donald R. Morrison, as *radix trees* são úteis para construir estruturas associativas com chaves que podem ser expressas como cadeias de caracteres. Essas características tornam as mesmas úteis para aplicações na área de roteamento IP, onde a capacidade de conter grandes intervalos de valores com poucas exceções é particularmente adequada para a organização hierárquica de endereços.

No projeto aqui descrito, a estrutura de dados *radix trie* é implementada utilizando o pacote *triebeard* [47].

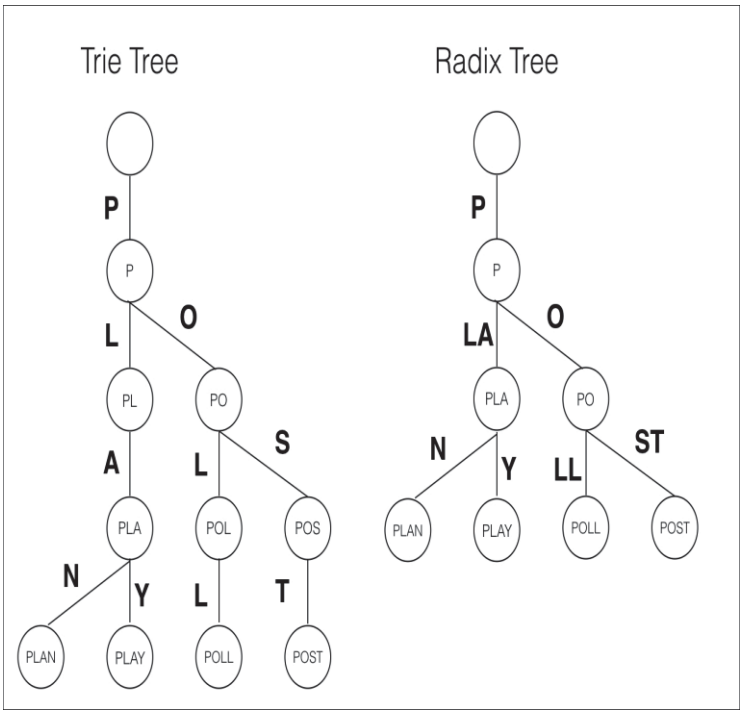

Figura 11. *Trie tree* (esqueda) e *radix tree* (direita) com as mesmas entradas [48].

4.8. RIPE ATLAS

A plataforma *Atlas* é uma das maiores redes de monitoramento da Internet. Com ela é possível monitorar de forma contínua a acessibilidade da rede a partir de milhares de pontos espalhados por todo o globo terrestre, investigar de forma rápida problemas de conexões, criar alarmes usando as verificações de status do *RIPE Atlas*, que funcionam com suas próprias

ferramentas de monitoramento, verificar a capacidade de resposta das infraestruturas de *DNS* e realizar testes de conectividade IPv6. A coleta de dados do *RIPE Atlas* se dá por meio de dois dispositivos chamados *probes e anchors*, apresentados nas Figuras 12 e 13, respectivamente.

As *probes* são pequenos dispositivos de hardware, alimentados via *USB*, que se conectam à Internet por uma porta *Ethernet* e realizam diferentes medições, transmitindo os dados para a *RIPE NCC*, agregando-os com o restante dos dados da rede *RIPE Atlas*. As medições conduzidas pelos mesmos são: *ping, traceroute, SSL / TLS, DNS, NTP e HTTP* (para alvos seletivos). Os *anchors* do RIPE Atlas são *probes* aprimoradas com uma maior capacidade de medição, bem como potentes alvos de medição regionais. Até o primeiro semestre de 2018 haviam por volta de 10400 *probes* e 320 *anchors* conectados à Internet [49].

Figura 12. *Probe RIPE Atlas* [49]

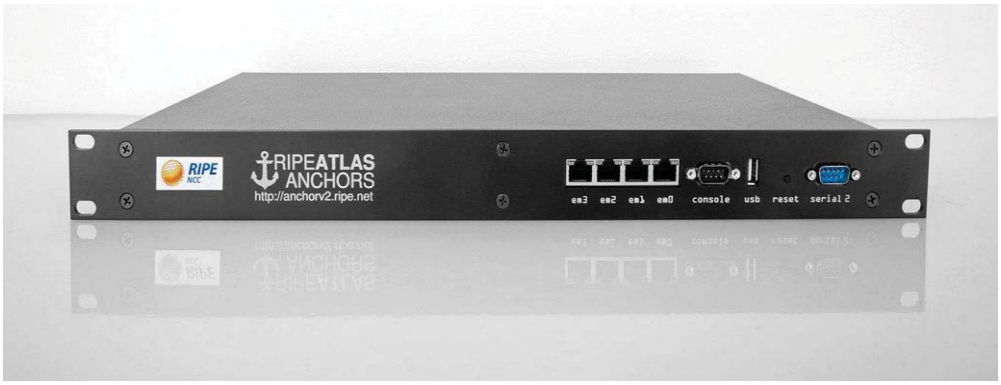

Figura 13. *Anchor RIPE Atlas* [49].

Além da possibilidade de programar experimentos na *RIPE Atlas*, existem alguns monitoramentos, chamados de *built-in measurements*[14], que são executados de forma contínua e automática pelas *probes*. *Ping*, *traceroute*, *DNS*, *SSL*/*TLS* e alguns tipos de *HTTP*,

[14] https://atlas.ripe.net/docs/built-in/

principalmente para destinos conhecidos, como servidores *Root Servers*, mas também para alguns dos componentes da infraestrutura *RIPE Atlas*, fazem parte do leque das medições contínuas existentes.

5. CATCHMENTVIEW

Intitulada *CatchmentView*[15], a ferramenta web apresentada neste trabalho foi projetada e desenvolvida com o objetivo auxiliar na identificação das diferenças entre o IPv4 e IPv6 *catchment* em serviços *anycast*. A ferramenta utiliza os resultados de medições *traceroute* como dados de entrada para exibir gráficos que auxiliam a inferir a topologia da rede e as características dos vários componentes envolvidos no roteamento.

Neste trabalho, a plataforma RIPE Atlas e os *Root Servers*, já mencionados anteriormente, foram utilizados como caso de estudo. Essa escolha se deu pelos *Root Servers* serem um dos maiores utilizadores de *anycast*, e pela plataforma RIPE Atlas ser uma das maiores redes mundiais de monitoramento da Internet. Dessa forma, a descrição da ferramenta, a ser realizada neste capítulo, utilizará de exemplos e termos presentes nessas tecnologias. Entretanto, a ferramenta não tem seu uso atrelado exclusivamente as mesmas. Outras bases de dados podem ser utilizadas e outros serviços que utilizem *anycast* como os *Content Delivery Networks - CDN*, fornecidos por *grandes companhias como Akamai, Amazon's AWS, Microsoft Azure* entre outras, podem usufruir da ferramenta no intuito de averiguar as diferenças das requisições realizadas em IPv4 e IPv6 para seus serviços.

5.1. OVERVIEW

A ferramenta de visualização é formada por dois módulos denominados *IPv4 vs IPv6 e Temporal*. O primeiro tem por objetivo demonstrar graficamente as diferenças entre o *catchment IPv4 e IPv6* do serviço *anycast* selecionado. Dessa forma, apenas os dados das medições que alcançaram o destino em ambas as versões do IP são considerados e dois gráficos, dispostos lado a lado, são exibidos. Já no módulo *Temporal*, apenas um único gráfico é exibido, correspondente às medições IPv6 ou IPv4. Esse módulo tem por objetivo prover uma maior ênfase à análise de uma das versões do IP.

As informações mínimas a serem utilizadas para definir os gráficos que serão exibidos na ferramenta são: o serviço *anycast*, a data e hora em que a medição ocorreu e a versão do IP para as visualizações temporais. Um exemplo de definição de um gráfico seria configurar a ferramenta para o *Root Server* K, as 12 horas do dia 14 de agosto de 2018 e

[15] CatchmentView. Disponível em: http://catchmentview.com/

selecionar a versão do IP (v6 ou v4) se o módulo desejado for o *Temporal*. Nesta configuração, a ferramenta disponibilizará informações de todas as *probes* disponíveis, seguindo os critérios de serviço, data, hora e versão do IP selecionados.

Além disso, as seguintes configurações avançadas de filtro podem ser utilizadas para definir as *probes* que farão parte dos gráficos:

- *Probes ID:* O usuário pode configurar para que os gráficos sejam compostos somente pelas *probes* selecionadas por ele. Por Exemplo, o usuário pode definir as *probes* de ID 8, 9 e 10 como entrada. Dessa forma, somente as informações das requisições enviadas por essas *probes* irão compor o gráfico;
- *AS number:* O usuário pode analisar as medições provenientes de *probe* situadas em determinados *autonomous systems.* Para isso, basta informar como entrada os *ASN* desejados. Por exemplo: se o usuário definir como entrada os *autonomous systems* 7786 e 6993, apenas as medições das *probes* situadas nestes *autonomous systems* serão levadas em consideração;
- Região do globo: Visualizar o comportamento do roteamento por meio de requisições oriundas de *probes* situadas em determinada parte do globo pode ser a necessidade de algum operador. Dessa forma, a ferramenta oferece a possibilidade dos usuários selecionarem uma das 5 regiões do globo: *West, North Central, South Central, North East, South East*. Essa divisão pode ser conferida na Figura 14 e foi determinada pela *RIPE Atlas*, a qual utiliza desta divisão em sua plataforma de medições;
- País: O usuário pode selecionar as *probes* que farão parte de sua análise selecionando o país em que elas se encontram. Por exemplo, se um usuário selecionar Brasil, Itália e Estados Unidos, somente as *probes* que se encontram nesses países serão consideradas.

Essas definições avançadas de seleção são opcionais e acumulativas. Sendo assim, um usuário pode utilizar mais de uma configuração para filtrar as *probes* que farão parte da análise. Selecionar determinados *AS* e utilizar o filtro de países, pode ser um exemplo válido nesse caso.

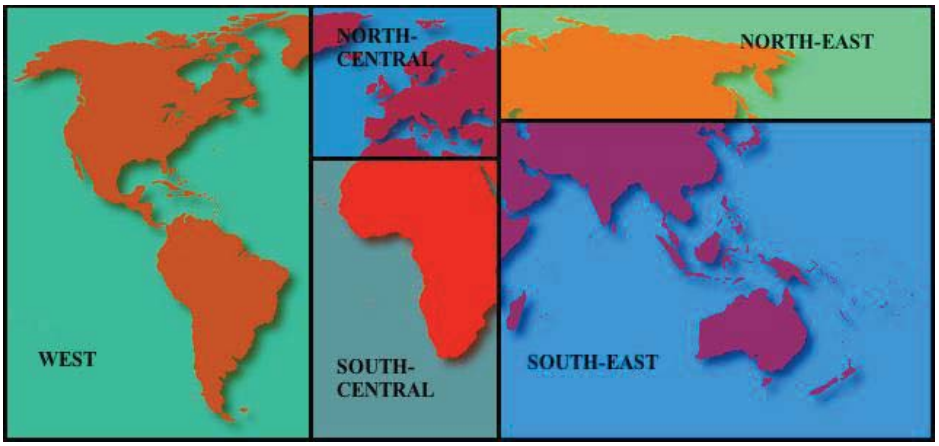

Figura 14. Áreas utilizadas nas fontes de medições da plataforma RIPE Atlas [50].

Na Figura 15 é apresentada uma visão geral da ferramenta utilizando o módulo *IPv4 vs IPv6*. Na parte esquerda da ferramenta encontram-se as opções de seleção do gráfico, na parte superior 3 *valueboxes* com informações e interações referentes as medições são exibidos e por fim, na parte central da ferramenta os gráficos que representam o c*atchment* são apresentados.

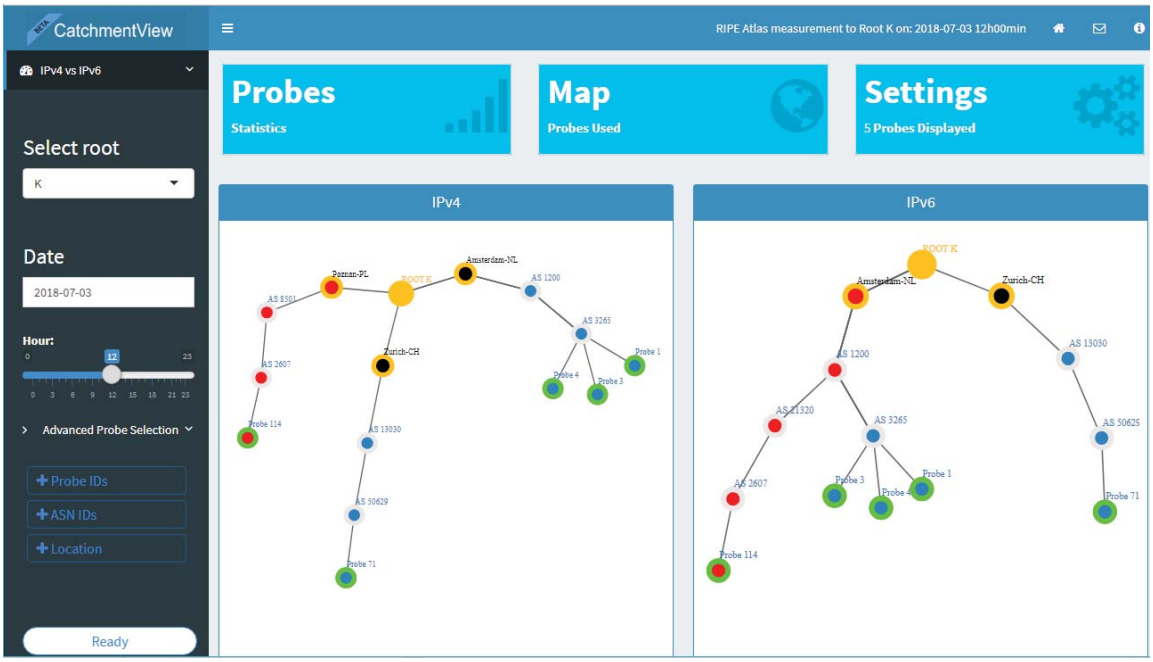

Figura 15. Visão geral da ferramenta.

Os resultados das medições que compõem o *catchment*, são apresentados em um *force-directed-graph* formado por nodos e vértices. Na tabela 2 são apresentados os nodos presentes nos gráficos e uma breve descrição do seu significado na aplicação. Os nodos de cor laranja, situados na região central do gráfico, correspondem ao *target* conceitual. Os nodos

pretos com bordas laranjas, conectados ao *target*, representam as prováveis cópias do serviço *anycast* que receberam as requisições. Os nodos internos azuis com bordas cinzas representam os *autonomous systems* em que a requisição trafegou até alcançar ao *target*. Por fim, os nodos azuis com bordas verdes, localizados geralmente na periferia do gráfico, representam as *Probe RIPE Atlas.*

Tabela 2. Nodos presentes no gráfico e seu significado para a aplicação

Nodo	Significado
	*Target* conceitual
	*Mirror*
	*Autonomous System*
	*Probe RIPE Atlas*

5.2. DADOS DE ENTRADA

A ferramenta utiliza dados oriundos de resultados de medições do tipo *traceroute*, disparadas das Probes RIPE Atlas para os *Root Servers*. Essas medições, denominadas *traceroute built-in*[16], são enviadas periodicamente a cada 30 minutos por todas *probes* da RIPE Atlas conectadas a Internet, utilizando o protocolo *UDP (User Datagram Protocol)* com 3 pacotes encaminhados, tempo limite de resposta de 1000 milissegundos e *hops* ilimitados. Na tabela 3, são exibidas informações sobre a medição *traceroute* enviada periodicamente pela plataforma *RIPE Atlas* aos *Root Servers*

Os resultados das medições são disponibilizados no formato *JSON (JavaScript Object Notation)* pela plataforma *RIPE Atlas* por meio de uma API. Com essa API é possível selecionar pelo ID a medição a ser analisada e selecionar o intervalo de tempo mediante a inserção dos *timestamp*[17] iniciais e finais.

Tabela 3. *Build-in traceroute* para os *Root Servers*

Destino	ID IPv4	ID IPv6

[16] Informações completas: https://atlas.ripe.net/api/v2/measurements/ + ID.
[17] Método utilizado na computação para rastrear o tempo. Iniciado em 1º de Janeiro de 1970, o *timestamp* é calculado pelo número de segundos existentes entre uma data específica e a data de início, conhecida como *Unix Epoch*.

a.root-servers.net	5009	6009
b.root-servers.net	5010	6010
c.root-servers.net	5011	6011
d.root-servers.net	5012	6012
e.root-servers.net	5013	6013
f.root-servers.net	5004	6004
g.root-servers.net	5014	6014
h.root-servers.net	5015	6015
i.root-servers.net	5005	6005
j.root-servers.net	5016	6016
k.root-servers.net	5001	6001
l.root-servers.net	5008	6008
m.root-servers.net	5006	6006

Na Figura 16, é apresentado um exemplo de uma medição *traceroute* gerada pela *probe* identificada pelo número 1. A descrição completa dos atributos existentes em cada medição pode ser conferida na *documentação*[18] das medições da plataforma *Atlas.*

[18] Disponível em: https://atlas.ripe.net/docs/data_struct/#v4750_traceroute

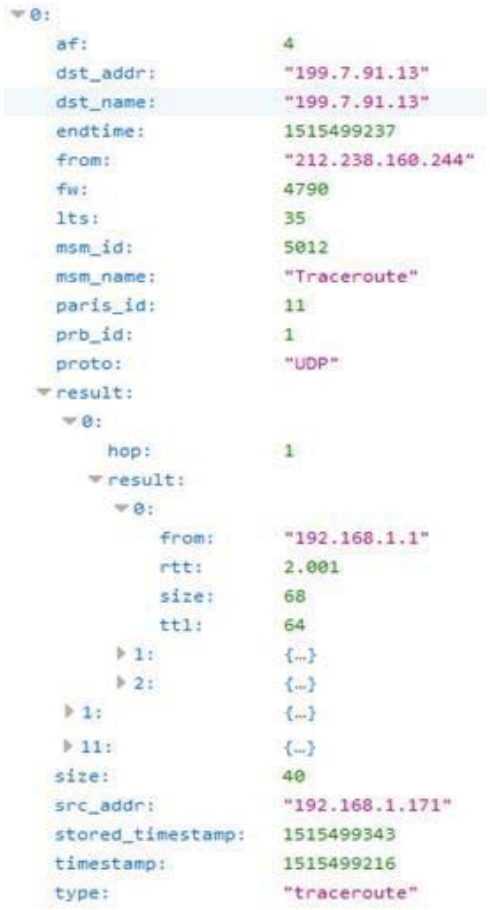

Figura 16. Exemplo de como os dados resultantes de uma medição *traceroute* são organizados, armazenados e fornecidos pela *RIPE Atlas*.

5.2.1. Servidor

Para a utilização das medições realizadas pela *RIPE Atlas*, desenvolveu-se um servidor responsável por coletar, filtrar, processar e armazenar esses dados. Implementado utilizando o ambiente e linguagem R, o servidor, a cada 60 minutos, realiza o *download* dos dados e inicia o processo de filtragem das informações. Nesse processo, as medições que não alcançaram o destino são descartadas, e nas demais os dados são processados a fim de gerar um novo arquivo com as medições no formato desejado, conforme ilustrado na Figura 17. Além disso, informações de localização e dos *AS* de cada *probe* são acrescentadas. Essas informações são obtidas na *lista diária de probes*[19], disponibilizada pela *RIPE Atlas*.

[19] Lista de probes, disponível em: https://ftp.ripe.net/ripe/atlas/probes/archive/

Figura 17. Formato do *JSON* de entrada.

As variáveis presentes em cada resultado representam informações referentes a *probe* que realizou aquela consulta, como: seu ID, o *ASN* da rede em que a *probe* está conectada, a latitude e longitude em que a *probe* se encontra e informações sobre o *traceroute* por ela realizado. As informações do *traceroute* são distribuídas em um objeto de nome *resultado*. Cada *hop* dado pela requisição é um novo objeto com o IP da máquina que partiu aquele salto, e o *RTT* mínimo e máximo, representados por *tmin* e *tmax*, respectivamente.

Os resultados filtrados são compactados e organizados seguindo a lógica mostrada na Figura 18. Divididos em pastas correspondentes ao ano/mês/dia em que as medições ocorreram, o nome dos arquivos é formado pelo nome do serviço *anyscast* seguido da palavra data, da versão do IP, de um subtraço (_) e do *timestamp* correspondente a data da medição.

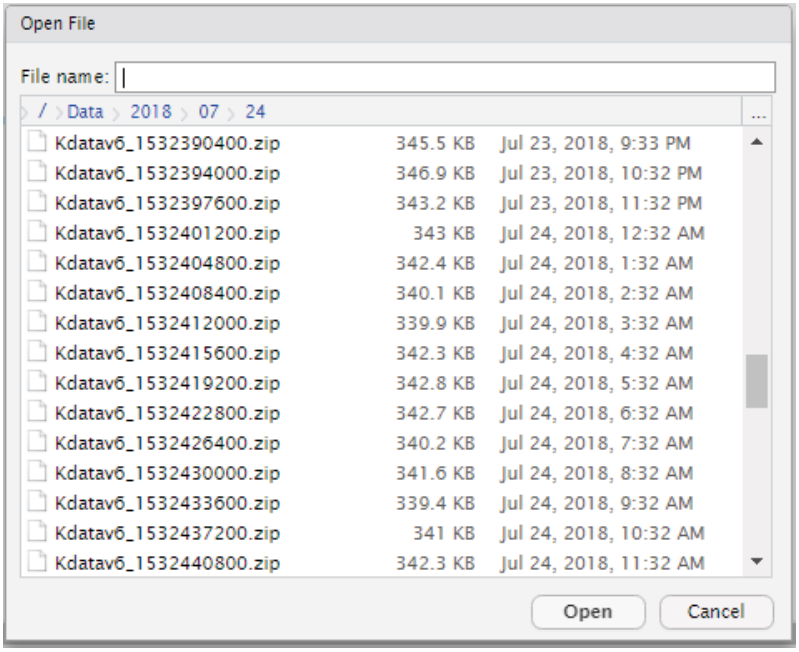

Figura 18. Organização dos arquivos filtrados pelo servidor.

5.3. FEATURES

Com o propósito de prover um maior entendimento das situações ilustradas e proporcionar maiores informações, a ferramenta disponibiliza as seguintes interações:

- *Highlight*: O usuário da plataforma pode destacar os nodos clicando nos mesmos. Ao fazer isso, o nodo selecionado receberá uma cor vermelha que o irá destacá-lo dos demais. Conforme demonstrado na Figura 15, se o nodo selecionado for uma *probe*, além dele, todo o caminho que o *traceroute* percorreu até o destino irá ser destacado. Também, se o módulo em questão for o *IPv4 x IPv6*, o caminho será destacado nos dois gráficos, afim de auxiliar a comparação dos *catchment;*
- *DoubleClick*: Clicando duas vezes em um nodo, o usuário tem acesso a uma série informações exibidas em *pop-ups*. De acordo com cada nodo, as seguintes informações são exibidas:
    - *Probe* nodos: Ilustrado na Figura 19, o *pop-up* exibido para os nodos que representam as *probes* contém informações gerais da mesma, como: seu *ID*, país e coordenadas geográficas, mapa com a sua localização e informações sobre a rede que está conectada. Além disso, uma tabela contendo informações sobre o *traceroute* por essa

probe encaminhado e a representação no mapa do caminho que os pacotes dessa requisição percorreram até chegar ao destino são exibidos;

- AS nodos: Nome do *Autonomous System holder* e contato, em alguns casos;
- *Mirror* nodos: Informações sobre a localização, *holder* e informações sobre a rede em que aquela cópia do serviço se encontra.

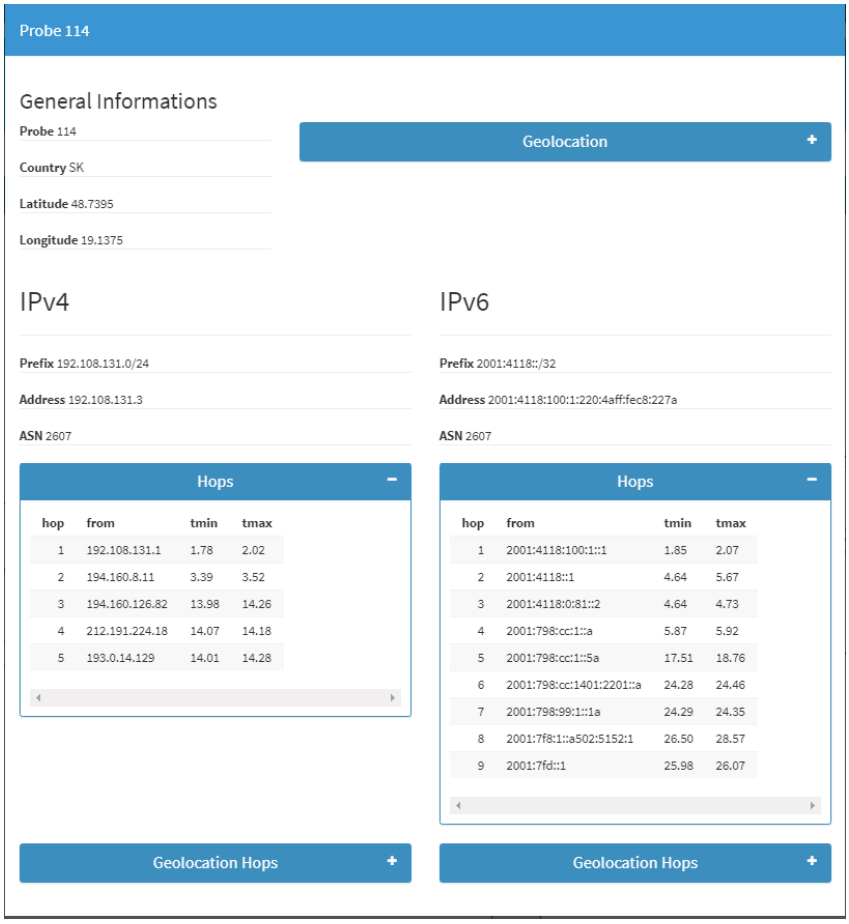

Figura 19. *Pop-up* de informações da *Probe.*

- *Value Boxes:* Três *value boxes* interativos são dispostos na parte de cima da ferramenta toda vez que um novo gráfico é apresentado. Na Figura 11 é possível observar os três *values boxes*, que possuem as seguintes funções:
  - *Statistics*: Esse *value box*, demonstrado na Figura 20, tem por objetivo demonstrar estatísticas gerais do *catchment* em questão;

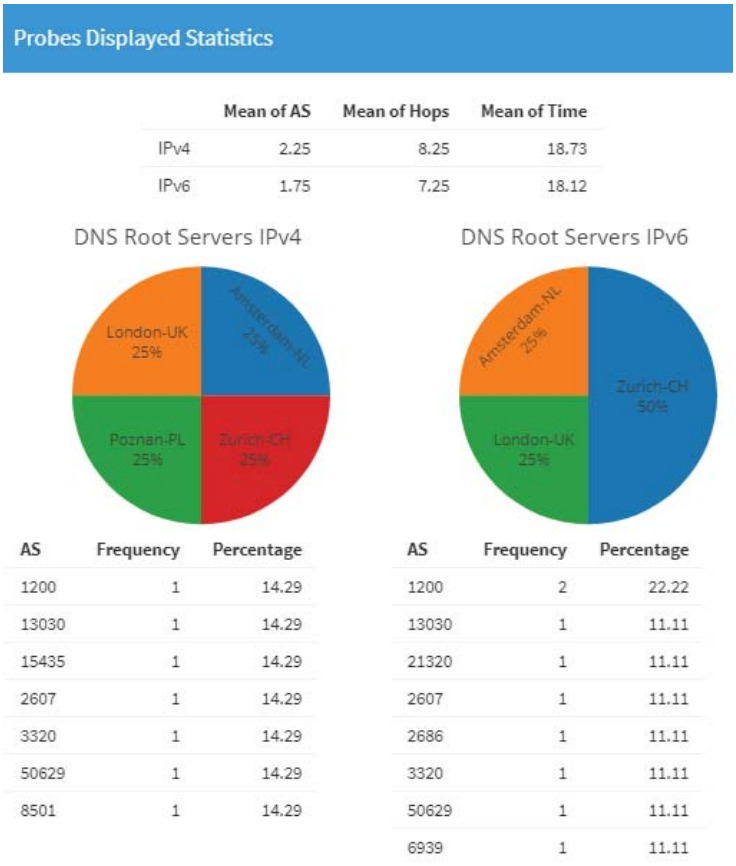

Figura 20. *Statistics pop-up.*

- *Map*: Apresenta em um mapa a localização das *probes* analisadas naquele *catchment*;
- *Settings*: O gráfico inicial é apresentado com apenas as 10 primeiras *probes* da medição solicitada pelo usuário. Entretanto, é possível mudar as *probes* que compõem o *catchment* ilustrado, clicando no *setting box* e selecionando as *probes* desejadas em uma lista de *probes* disponíveis. A Figura 21 apresenta um exemplo com a tabela de seleção da lista *probes*. Nesta tabela, as *probes* selecionadas possuem as suas linhas marcadas em tom azul e as não selecionadas em branco. Para marcar/desmarcar a linha, basta clicar em cima da mesma.

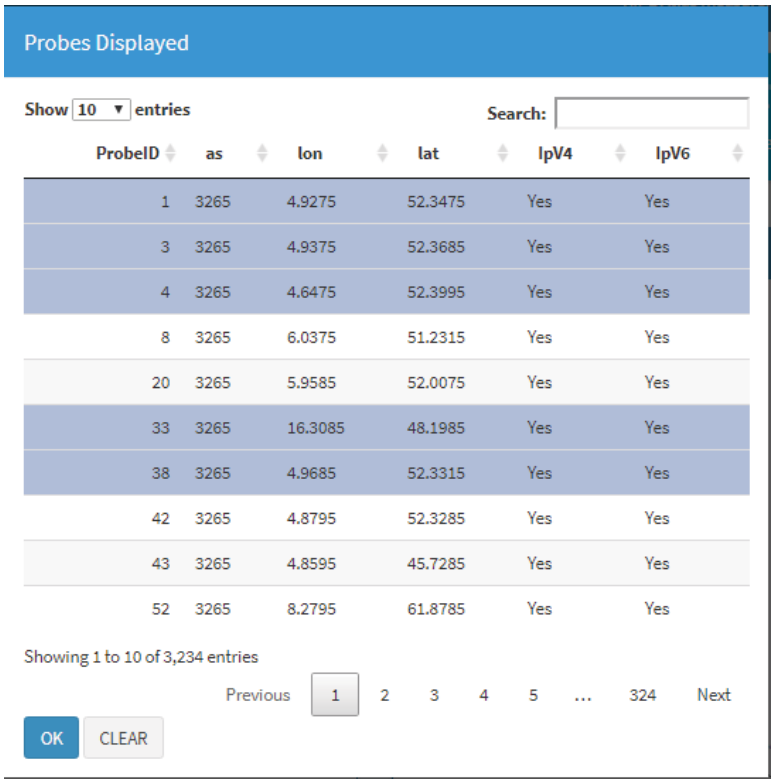

Figura 21. Tabela de seleção das *probes* exibidas no gráfico.

- Interações Nativas: Além das interações acima citadas, existem algumas interações nativas do gráfico *force-directed network*, como: *drag*, *zoom in* e *zoom out.*

5.4. TESTES E ANÁLISES

Durante o processo de implementação e testes, versões da ferramenta foram disponibilizadas a operadores do *Root Server B*. Esse contato foi fundamental na realização deste projeto, posto que, as situações e *feedback* apresentados por esses profissionais embasaram o desenvolvimento da ferramenta e reforçaram a real necessidade do monitoramento do *catchment* de serviços *anycast*, bem como das diferenças entre o IPv4 e IPv6. Dessa forma, pode-se dizer que a ferramenta se apresentou promissora quanto a sua utilização por parte dos operadores. Nesta sessão, alguns exemplos de uso da ferramenta, que reforçam a sua necessidade, serão apresentados.

Na Figura 22, um exemplo utilizando o módulo IPv4 vs IPv6 é demonstrado. A composição dos gráficos apresentados nesta figura utiliza o filtro avançado de *probes* para selecionar 5 *probes* situadas no Brasil que realizaram a medição *traceroute* no dia 16 de Agosto de 2018 às 17 horas (GMT - *Greenwich Mean Time*) em direção ao *Root Server* K. Os gráficos

que compõem a figura foram retirados da ferramenta e adaptadas para comparar o *catchment* nas versões IPv4 e IPv6. Dessa forma, a coloração presente na parte central dos nodos identifica qual cópia do *Root Server K* recebeu aquela requisição: em azul a cópia localizada em Buenos Aires – AR, em preto Montevideu – UY, em amarelo Londres – UK e em vermelho Miami – US ou Kansas City – US. Ainda, na tabela 4 algumas informações sobre os *catchment*, como a quantidade de *AS, hops e RTT em IPv4 e IPv6* das medições de cada *probe* são apresentados.

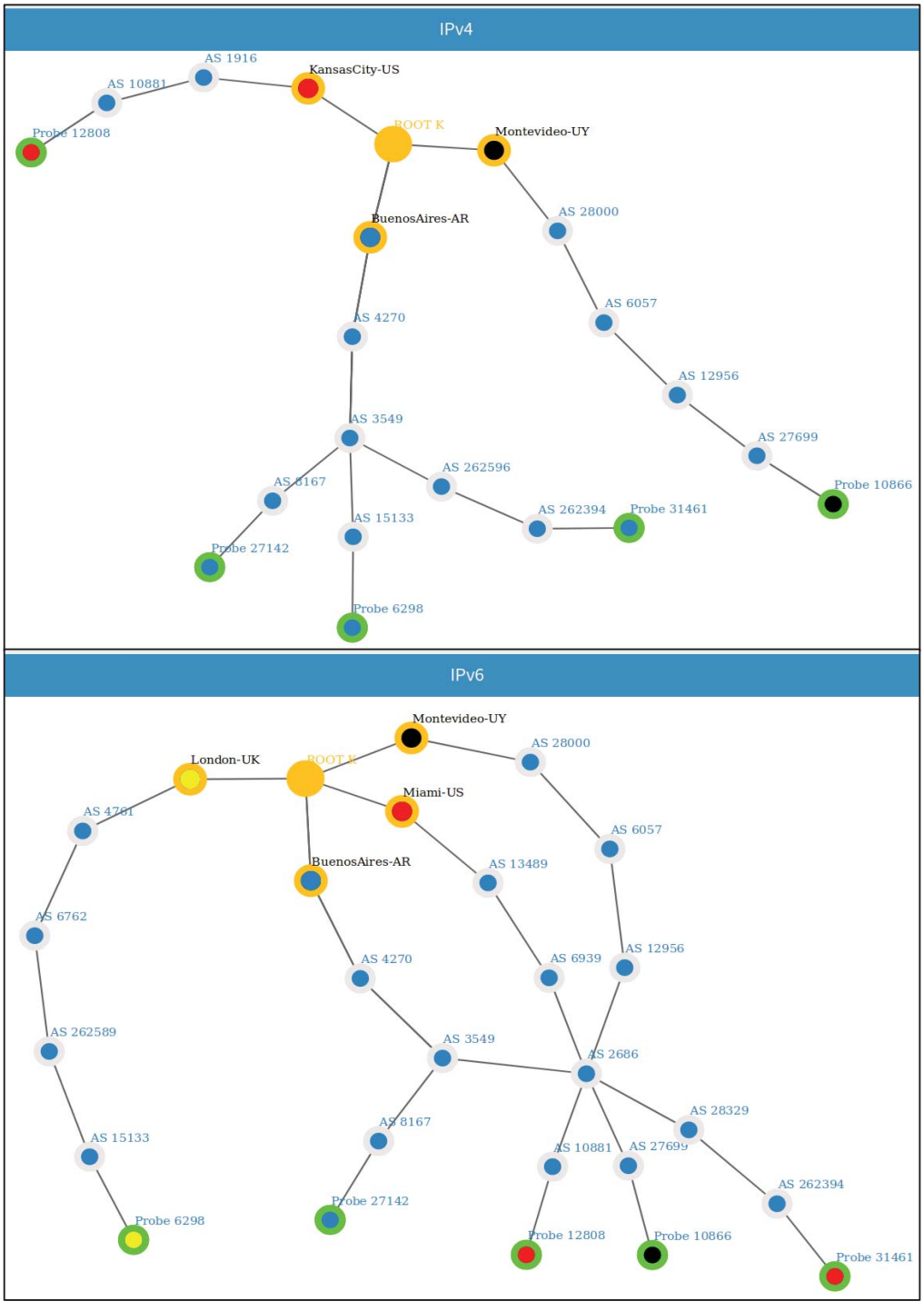

Figura 22. Exemplo de análise utilizando o módulo IPv4 vs IPv6.

Tabela 4. Resultado das medições

	IPv4		IPv6	
Probe	*AS*	*RTT*	*AS*	*RTT*
6298	3	28.5	4	244.2
10866	4	185.8	5	84.8
12808	2	123.3	4	139.9
27142	3	105.7	3	88.2
31461	4	57.8	5	152

Analisando os gráficos e as informações presentes na tabela, podemos constatar que nas medições das *probes* que alcançaram o mesmo destino em IPv4 e IPv6 (**27142 e 10866**), o *RTT* foi menor em IPv6, diferentemente das demais *probes* (**6298, 31461 e 12808**) que foram atendidas por diferentes cópias nas duas versões do IP e tiveram RTT maiores em IPv6. Essa diferença de desempenho se dá principalmente pelo caminho diferente que essas *probes* percorreram, além de trafegarem por um número maior de AS na versão 6 do IP. Sendo assim, essa análise enfatiza que caminhos diferentes em IPv6 estão contribuindo para uma menor performance das requisições realizadas nesta versão do IP, exatamente como Dhamdhere [17] e Wicaksana [29] constataram em seus trabalhos.

Outro exemplo, desta vez utilizando o módulo Temporal, é demonstrado na Figura 23. Nesta análise que utiliza as medições *traceroutes* IPv6 do dia 22 de Agosto de 2018 às 13 horas (GMT) com destino ao *Root Server K*, as *probes* **52, 93, 269 e 390** localizadas na Noruega, Nova Zelândia, Luxemburgo e Chile e situadas nos AS 29492, 681, 2602 e 27678, respectivamente, são utilizadas. Conforme ilustrado no gráfico, todas as medições enviadas por essas *probes* alcançaram diferentes cópias do *Root Server K*. O intrigante, é que todas as requisições passaram pelo AS 2686 antes de seguirem caminhos diferentes. Esse fato ocorre devido a políticas de roteamento impostas no protocolo BGP. Porém, essas definições devem ser de conhecimento dos operadores para que a qualidade dos serviços não seja afetada.

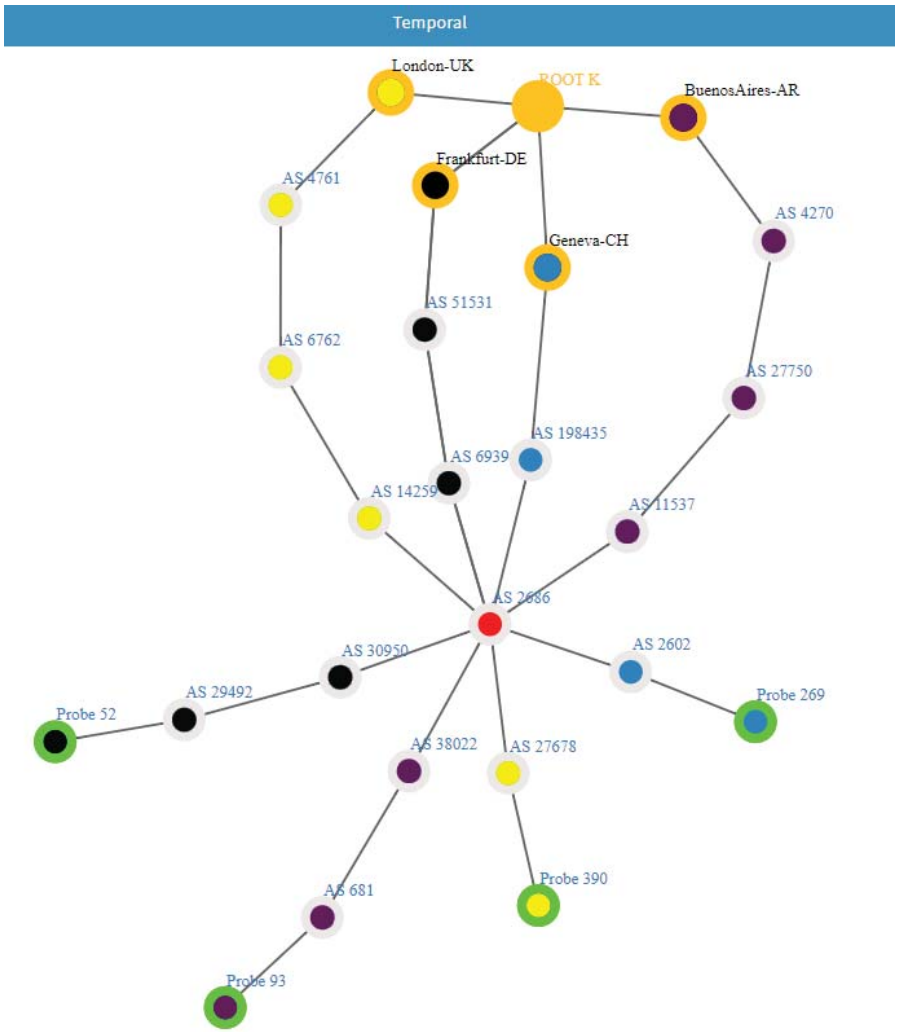

Figura 23. Exemplo de análise utilizando o módulo Temporal.

Comparar o desempenho de serviços *anycast* entre diferentes regiões do globo, com o objetivo de averiguar a eficiência dos serviços prestados em diferentes localidades, é também uma possibilidade de uso da ferramenta *catchmentView*. Um exemplo deste tipo de análise é demonstrado na Figura 24, a qual busca-se comparar o desempenho do Root Server K nas regiões: ***South-East, North-East, South-Central, North-Central e West*** (Figura 14). Para tal, utilizou-se o módulo IPv4 vs IPv6, com medições encaminhadas a esse serviço, no dia 21 de Agosto de 2018 às 14 horas (GMT). Além disso, com o filtro avançado de *probes*, delimitou-se a localização das *probes* nas regiões acima elencadas e selecionou-se apenas uma *probe* por AS de cada região. Dessa forma, na figura em questão, temos 5 tabelas extraídas da ferramenta, as quais demonstram a média de *AS, hops e RTT* de cada região.

South-East			
	Mean of AS	**Mean of Hops**	**Mean of Time**
IPv4	3.01	8.81	102.21
IPv6	3.34	9.09	128.21
North-East			
	Mean of AS	**Mean of Hops**	**Mean of Time**
IPv4	2.22	6.94	47.38
IPv6	3.33	9.06	89.99
South-Central			
	Mean of AS	**Mean of Hops**	**Mean of Time**
IPv4	1.92	6.96	65.30
IPv6	2.21	6.71	74.12
North-Central			
	Mean of AS	**Mean of Hops**	**Mean of Time**
IPv4	2.17	6.71	17.62
IPv6	2.84	6.67	17.29
West			
	Mean of AS	**Mean of Hops**	**Mean of Time**
IPv4	2.40	9.77	60.47
IPv6	3.46	10.32	74.16

Figura 24. IPv4 vs IPv6 em diferentes regiões do globo.

Analisando o tempo de consulta médio das regiões, é possível observar uma melhor eficiência das requisições enviadas por *probes* localizadas na região **North-Central**. O RTT médio de consulta das medições provenientes de *probes* localizadas nesta região foi de 17.62 em IPv4 e 17.29 em IPv6. Acredita-se, que grande parte dessa melhor eficiência se vale do fato de 55.56% dos servidores do Root Server K estarem localizados nesta região [21]. Outro ponto a se destacar é que a região ***North-Central***, foi a única possuir um menor RTT em requisições IPv6. Nas demais, as requisições em IPv6 tiveram um tempo significativamente maior, especialmente na região **North-East**, umas das regiões com menores adoções desta versão do IP [51] . No que diz respeito da média de AS utilizados, todas as regiões tiveram um maior valor médio em IPv6. Já a média de *hops* não seguiu visíveis padrões nesta análise, sendo maior

em IPv6 nas regiões ***South-East, North-East e West*** e menor nas regiões ***South-Central e North-Central.***

As análises demonstradas nesta sessão são apenas alguns dos exemplos de situações que podem ser observadas utilizando a ferramenta desenvolvida neste trabalho. Devido à complexidade de depurar o roteamento na Internet, em especial em serviços *anycast*, a utilização de ferramentas visuais, criadas para esse fim, torna-se fundamental para o conhecimento de acontecimentos, como os demonstrados, evitando que os mesmos passem despercebidos e afetem a qualidade dos serviços prestados.

6. CONSIDERAÇÕES FINAIS

A onipresença da Internet fez com que o número de dispositivos conectados crescesse de forma exponencial nos últimos anos. Esse crescimento, aliado a busca cada vez mais constante por resiliência, melhores performances, segurança e confiabilidade, impulsionaram uma exigência por significativas mudanças na Internet. O desenvolvimento e utilização de *anycast* por serviços de alta escalabilidade e a criação e adoção de uma nova versão do endereço IP, denominada IPv6, foram mudanças que ganharam destaque.

A implantação do IPv6 tornou-se necessária na medida que a quantidade de endereços IPv4 foi se esgotando. Entretanto, a transição entre as duas tecnologias vem ocorrendo forma lenta, fazendo com que os serviços disponibilizados na Internet tenham de operar em modo *Dual Stack*. Um dos problemas presentes nessa configuração, é a perda de qualidade dos serviços prestados em requisições onde há diferenças entre os caminhos percorridos em IPv4 e IPv6. Esse problema, acentua-se em serviços que utilizem *anycast* para a replicação de servidores, posto que, essa aplicação depende de um roteamento adequado para que seu propósito de buscar resiliência, melhores performances e maior confiabilidade sejam alcançados.

Diante da dificuldade de se identificar essas diferenças, do aumento da utilização de *anycast* e da estimativa de um longo caminho a ser percorrido para ser alcançada a total cobertura do IPv6, este trabalho teve como principal objetivo o desenvolvimento de uma ferramenta para auxiliar os operadores da Internet na compreensão das diferenças entre o IPv4 e IPv6 *catchment* em serviços *anycast*.

A ferramenta buscou ser direta e interativa, afim de transparecer de forma clara aos operadores situações que sem o auxílio da mesma passariam despercebidas ou despenderiam um tempo muito maior de análise e compreensão. Este trabalho, utilizou os *Root Servers* e a plataforma *RIPE Atlas* como caso de estudo. Essas duas tecnologias foram fundamentais no desenvolvimento e na validação do projeto, pois ambas auxiliaram a moldar as funcionalidades da ferramenta. Entretanto, como já foi destacado, a ferramenta não está atrelada diretamente a essas duas tecnologias, podendo ser utilizada com outras bases de dados livres ou proprietárias e para outros serviços que utilizem *anycast*.

Após os testes e validações, compreende-se que a ferramenta desenvolvida pode ser utilizada como uma aliada na compreensão das diferenças entre o IPv4 e IPv6 *catchment* de serviços *anycast*, podendo assim, auxiliar na melhoria dos serviços prestados. Também, por

meio dos testes, verificou-se a necessidade de algumas modificações no projeto. Assim sendo, como trabalho futuro, estuda-se ampliar a precisão na descoberta das cópias dos serviços que receberam as requisições, por meio de banco de dados mais precisos ou com o acréscimo de maiores informações provenientes de outras medições, como *queries DNS*, por exemplo. Além disso, prevê-se o acréscimo de novas funcionalidades que automatizem algumas comparações, de forma a tornar mais rápida a análise por parte dos operadores.

REFERÊNCIAS

[1] PRESS, G. Internet Of Things By The Numbers: What New Surveys Found. Disponível em:<https://www.forbes.com/sites/gilpress/2016/09/02/internet-of-things-by-the-numbers-what-new-surveys-found/#5490885d16a0>. Acesso em: 26 Set. 2017.

[2] MILLIKEN, W.; MENDEZ, T.; PARTRIDGE, C. RFC 1546: Host Anycasting Service. Disponível em: <https://tools.ietf.org/html/rfc1546>. Acesso em: 27 Set. 2017.

[3] MATTHEW PRINCE. A Brief Primer on Anycast. Disponível em: <https://blog.cloudflare.com/a-brief-anycast-primer/#comment-879815852>. Acesso em: 17 Set. 2017.

[4] COLITTI, L. *et al.* Evaluating the effects of anycast on DNS root name servers. 2006. Disponível em: <https://www.ripe.net/publications/docs/ripe-393>. Acesso em: 15 Set. 2017.

[5] BHATTACHARJEE, S. *et al.* Application-layer anycasting. *Proceedings of INFOCOM '97*. vol. 3. p. 1388–1396.

[6] WEBER, S.; CHENG, L. A survey of anycast in IPv6 networks. *IEEE Commun. Mag.* vol. 42, no. 1. p. 127–132. jan. 2004.

[7] CAESAR, M.; REXFORD, J. BGP routing policies in ISP networks. *IEEE Netw.* vol. 19, no. 6. p. 5–11. 2005.

[8] POSTEL, J. RFC 791: Internet Protocol. Disponível em: <https://tools.ietf.org/html/rfc791>. Acesso em: 27 Set. 2017.

[9] KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet : uma abordagem top-down*. 6th ed. Pearson, 2013.

[10] Routing (Data Communications and Networking). Disponível em: <http://what-when-how.com/data-communications-and-networking/routing-data-communications-and-networking/>. Acesso em: 10 Ago. 2018.

[11] HARES, S.; REKHTER, Y.; LI, T. A Border Gateway Protocol 4 (BGP-4). Disponível em: <https://tools.ietf.org/html/rfc4271>. Acesso em: 28 Set. 2017.

[12] CISCO. Estudos de caso de BGP. Disponível em: <https://www.cisco.com/c/pt_br/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html#sec2>. Acesso em: 04 Out. 2017.

[13] TANENBAUM, A. S. *Computer Networks*. 4th ed., vol. 291. Prentice Hall PTR, 2012.

[14] RIPE. IPv6 Enabled Networks. Disponível em: <http://v6asns.ripe.net/v/6?s=_ALL>. Acesso em: 04 Out. 2017.

[15] SRISURESH P, E. K. RFC 3022: Traditional IP Network Address Translator (Traditional NAT). Disponível em: <https://www.ietf.org/rfc/rfc3022.txt>. Acesso em: 04 Out. 2017.

[16] NIKKHAH, M.; GUERIN, R. Migrating the Internet to IPv6: An Exploration of the When and Why. *IEEE/ACM Trans. Netw.* vol. 24, no. 4. p. 2291–2304. ago. 2016.

[17] DHAMDHERE, A. *et al.* Measuring the deployment of IPv6. *Proceedings of the 2012 ACM conference on Internet measurement conference - IMC '12*. 2012. p. 537.

[18] FLOODSITE. Catchment area. Disponível em: <http://www.floodsite.net/juniorfloodsite/html/en/student/thingstoknow/hydrology/catchmentarea.html>. Acesso em: 18 Set. 2018.

[19] MOCKAPETRIS, P. RFC 1035: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. Disponível em: <https://www.ietf.org/rfc/rfc1035.txt>. Acesso em: 04 Out. 2017.

[20] ISC. BIND Open Source DNS Server | Internet Systems Consortium. Disponível em: <https://www.isc.org/downloads/bind/>. Acesso em: 04 Out. 2017.
[21] Root Server Technical Operations Assn. Disponível em: <http://www.root-servers.org/>. Acesso em: 04 Out. 2017.
[22] IANA. Root Zone Database. Disponível em: <https://www.iana.org/domains/root/db>. Acesso em: 04 Out. 2017.
[23] P.VIXIE,G.SNEERINGER, M. S. Events of 21-Oct-2002. Disponível em: <http://c.root-servers.org/october21.txt>. Acesso em: 04 Out. 2017.
[24] ICANN. Root server attack on 6 February 2007. *N. S. W. Public Health Bull.* no. February. p. 6. 2007.
[25] FAN, X.; HEIDEMANN, J.; GOVINDAN, R. Evaluating anycast in the domain name system. *Proceedings - IEEE INFOCOM*. 2013. p. 1681–1689.
[26] BARTOLOMEO, M. Di. Visual Analytics of Network Routing Through Traceroute Data: Models and Techniques.2016. 175p.Tese (Doutorado). Computer Science and Automation, Roma Tre University, Roma.
[27] CANDELA, M. *et al.* Dynamic Traceroute Visualization at Multiple Abstraction Levels. 2013. Disponível em: <http://link.springer.com/10.1007/978-3-319-03841-4_43>. Acesso em: 18 Set. 2018.
[28] MASSIMO CANDELA. TraceMON: Network Debugging Made Easy — RIPE Labs. Disponível em: <https://labs.ripe.net/Members/massimo_candela/tracemon-traceroute-visualisation-network-debugging-tool>. Acesso em: 12 Jul. 2018.
[29] WICAKSANA, M. A. IPv4 vs IPv6 anycast catchment: A Root DNS study. Dissertação (Mestrado). Faculty of Electrical Engineering, Mathematics and Computer Science - EEMCS University of Twente. 2016.
[30] IHAKA, R.; GENTLEMAN, R. R: A Language for Data Analysis and Graphics. *J. Comput. Graph. Stat.* vol. 5, no. 3. p. 299–314. set. 1996.
[31] R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Disponível em: <https://www.R-project.org/>.
[32] RESNIZKY, H. G. Learning Shiny: Make the Most of R's Dynamic Capabilities and Create Web Applications With Shiny. 2015. p. 5.
[33] CHANG, W.; Borges Ribeiro, B.shinydashboard: Create Dashboards with 'Shiny'. [S.l.], 2017. R package version 0.6.1. Disponível em: <https://CRAN.R-project.org/package=shinydashboard>.
[34] KOBOUROV, S. G. Spring Embedders and Force Directed Graph Drawing Algorithms. jan. 2012.
[35] TUTTE, W. T. How to Draw a Graph. *Proc. London Math. Soc.* vol. s3-13, no. 1. p. 743–767. 1963.
[36] EADES; P. A heuristic for graph drawing. *Congr. Numer.* vol. 42. p. 149–160. 1984.
[37] TEJA, S. C.; YEMULA, P. K. Power network layout generation using force directed graph technique. *2014 18th National Power Systems Conference, NPSC 2014*. 2015.
[38] BANNISTER, M. J. *et al.* Force-directed graph drawing using social gravity and scaling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2013. vol. 7704 LNCS. p. 414–425.
[39] ALLAIRE, J. et al. networkD3: D3 JavaScript Network Graphs from R. [S.l.], 2017. R package version 0.4. Disponível em: <https://CRAN.R-project.org/package=networkD3>.
[40] MAXMIND. GeoIP2 City. Disponível em: <https://www.maxmind.com/pt/geoip2-city-accuracy-comparison>. Acesso em: 30 Ago. 2018.
[41] JAMES INMAN. *Navigation and Nautical Astronomy, for the Use of British Seamen*. F.

& J. Rivington, 1949.

[42] BRUMMELEN, G. Van. *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. Princeton University Press, 2012.

[43] DE MENDOZA Y RÍOS, J. *Memoria sobre algunos metodos nuevos de calcular la longitud por las distancias lunares y explicaciones prácticas de una teoría para la solución de otros problemas de navegación*. Imp. Real, 1795.

[44] NORDIN, N. A. M. *et al.* Finding shortest path of the ambulance routing: Interface of A* algorithm using C# programming. *2012 IEEE Symposium on Humanities, Science and Engineering Research*. 2012. p. 1569–1573.

[45] HIJMANS, R. J. geosphere: Spherical Trigonometry. 2017. R package version 1.5-7. Disponível em: <https://CRAN.R-project.org/package=geosphere>

[46] MORRISON, D. R.; R., D. PATRICIA---Practical Algorithm To Retrieve Information Coded in Alphanumeric. *J. ACM*. vol. 15, no. 4. p. 514–534. out. 1968.

[47] KEYES, O.; SCHMIDT, D.; TAKANO, Y.triebeard: 'Radix' Trees in 'Rcpp'. [S.l.], 2016. R package version 0.3.0. Disponível em: <https://CRAN.R-project.org/package=triebeard>.

[48] ALEBICTO, M. E.; AZAR, E. Radix tree. in *Swift Data Structure and Algorithms*. Packt Publishing. 2016. p. 200,209.

[49] RIPE Atlas - RIPE Network Coordination Centre. Disponível em: <https://atlas.ripe.net/>. Acesso em: 04 Out. 2017.

[50] RIPE Atlas - User-Defined Measurements - RIPE Atlas — RIPE Network Coordination Centre. Disponível em: <https://atlas.ripe.net/docs/udm/>. Acesso em: 19 Jul. 2018.

[51] GOOGLE. Statistics about IPv6 adoption. Disponível em: <https://www.google.com/intl/en/ipv6/statistics.html#tab=per-country-ipv6-adoption>. Acesso em: 14 Set. 2018.