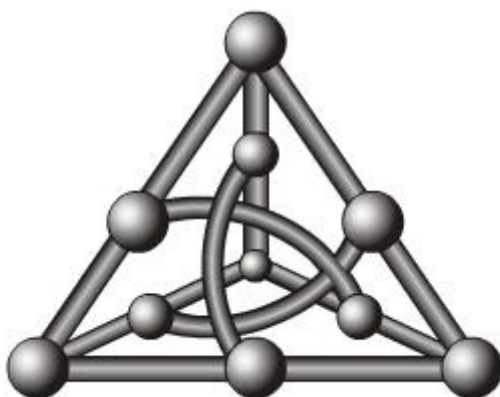

Controle no roteamento de rotas utilizando o protocolo BGP

Gilson Gabriel Zozias de Santana

**Projeto de graduação apresentado ao
Departamento de Computação e Estatística da
Universidade Federal de Mato Grosso do Sul, para
obtenção do grau de Bacharel em Ciência da Computação**



Orientador: Prof. Ronaldo Alves Ferreira

Campo Grande, Julho de 2018

Resumo

A Internet transformou-se, ao longo dos anos, em um dos meios tecnológicos mais disseminados mundialmente. Onde os provedores de serviços de internet(ISP) começaram a modificar as configurações de roteamento para apoiar políticas de roteamento, no qual o proprietário do roteador que controlava quais as rotas foram escolhidas e quais rotas foram propagadas para os vizinhos. Portanto esta proposta usa um protocolo de arquitetura de rede usando o Border Gateway Protocol(BGP) que atende aos requisitos para conectar, monitorar e analisar o tráfego entre os clientes desses provedores. Este artigo apresenta um estudo dos aspectos que envolvem o roteamento usando o protocolo BGP, como os seus recursos, e escolhas de decisões econômicas e políticas envolvidas no roteamento. O resultado deste trabalho foi usado na implantação de um ambiente, onde podemos monitorar o operadores fornecendo os fluxos de dados.

Sumário

1	Introdução	1
2	Roteamento Interno	3
2.1	Algoritmo do caminho mais curto	4
2.2	Interfaces OSPF	4
2.3	Áreas OSPF	5
2.4	Vantagens do protocolo OSPF	6
3	Protocolo BGP	7
3.1	BGP Interno	8
3.2	BGP Externo	10
3.3	Critérios na seleção de rotas	10
3.4	Importância do uso das políticas BGP	11
4	Projeto e a sua implementação	13
5	Avaliação da Implementação	17
6	Conclusão	22

Capítulo 1

Introdução

A definição simples de Internet é que representa uma coleção de redes interconectadas, enquanto os roteadores podem ser definidos, como a intersecção que liga essas redes, ou seja, os pontos que possibilitam essa ponte. Estes, por sua vez, estão organizados de forma hierárquica, onde alguns roteadores são utilizados para trocar dados entre grupos de redes controlados pelo mesmo domínio, enquanto outros roteadores fazem também a comunicação entre domínios. Um grupo de redes IP, sobre uma gerência técnica e que compartilham uma mesma política de roteamento se chama sistema autônomo (AS).

O roteamento é a forma mais importante para a entrega de pacotes de dados entre roteadores na internet, através de uma infraestrutura de redes interconectadas. Para tanto, o roteamento executa o processamento de rotas para um determinado sistema de comunicação, em que são necessários dois elementos: tabelas de roteamento e protocolos de roteamento. As tabelas de roteamento são registros de endereços de destino associados as métricas para alcançar esse destino, podendo conter outras informações. Os protocolos de roteamento determinam os conteúdos das tabelas de roteamento, ou seja, eles ditam como a tabela é montada e com quais informações ela é composta. Existem dois tipos de algoritmos atualmente em uso, pelos protocolos de roteamento: o algoritmo baseado em Vetor de Distância (*Distance-Vector Routing Protocols*) e o algoritmo baseado no Estado de Enlace (*Link State Routing Protocols*).

Todos os protocolos de roteamento realizam as mesmas funções básicas. Eles determinam a melhor rota para cada destino e distribuem as informações de roteamento entre os sistemas da rede. A forma pela qual é decidida a melhor rota é o que determina a diferença entre os protocolos de roteamento existentes, que podem ser internos ou externos. No protocolo de roteamento interno, os roteadores que são utilizados para trocar informações dentro do sistemas autônomos, são chamados de roteadores internos (*internal routers*) e podem usar uma variedade de protocolos de roteamento interno (*Interior Gateway Protocols - IGP*s). Dentre eles estão: RIP, IGRP, EIGRP, OSPF, sendo esses últimos os mais usuais. Já no protocolo de roteamento externo, os roteadores que trocam dados entre sistemas autônomos são chamados de roteadores externos (*external routers*), no qual utilizam protocolos tal como o BGP (*Border Gateway Protocol*).

Para este tipo de roteamento são considerados basicamente coleções de prefixos CIDR (*Classless Inter Domain Routing*) identificados pelo número de um sistema autônomo.

Antigamente, o caminho mais curto do roteamento era tipicamente usado. Ao longo do tempo, como a internet tornou-se mais comercializada e privatizada, prestadores de serviços de internet (*ISPs*) começaram a ter interesses em controlar o tráfego de dados por razões econômicas e políticas. Com isso o protocolo BGP nasceu da necessidade dos ISPs de controlar a seleção e a propagação de rotas. O resultado é um protocolo em que a maioria da complexidade está no processo de decisão e as políticas utilizadas para influenciar nas decisões.

O primeiro objetivo deste projeto é estudar detalhadamente o que acontece com o funcionamento do protocolo BGP. Com isto, será feita uma análise do ambiente de redes visando avaliar os comportamentos possíveis, suas causas e estratégias. Para demonstrar as variedades de técnicas a fim de fornecer o controle de roteamento do protocolo BGP, foi realizada uma simulação baseada em um ambiente real, com quatro backbones: RNP (Rede Nacional de Ensino e Pesquisa), EBT (Embratel), I2 (Internet Comercial 2), BF (Backbone Final). Onde temos relacionamentos entre eles vistos na Figura 1.1. A implementação do ambiente foi realizado no roteador de software BIRD para o plano de controle e o Kernel Linux para o plano de dados. O Capítulo 2 faz uma análise do protocolo de roteamento interno usado no projeto. No Capítulo 3 traz uma análise do protocolo BGP, tal como o processo de seleção de rotas, importação e exportação de rotas, e detalha a importância das políticas no protocolo BGP. No Capítulo 4 mostraremos o projeto e a sua implementação. O Capítulo 5 apresenta a avaliação da implementação do trabalho. Finalmente, o Capítulo 6 traz a conclusão deste projeto.

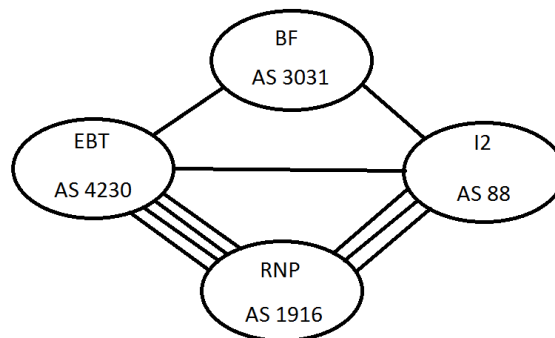


Figura 1.1: Ambiente do projeto simulado, com o relacionamento de quatro sistemas autônomos, e seus respectivos números

Capítulo 2

Roteamento Interno

Neste capítulo do trabalho são abordados conhecimentos básicos do protocolo de roteamento interno utilizado no projeto.

Um protocolo de roteamento intra-AS é usado para determinar como é realizado o encaminhamento de pacotes dentro de um sistema autônomo. Protocolos de roteamento intra-AS também são denominados como protocolos de roteadores internos IGP. Um AS agrupa roteadores que estão sob o mesmo controle administrativo. Ou seja, operados pelo mesmo ISP ou uma mesma corporação. Todos os roteadores do mesmo AS utilizam o mesmo algoritmo de roteamento, no caso deste projeto, o algoritmo utilizado é o OSPF (*Open Shortest Path First*).

O OSPF é um protocolo de roteamento dinâmico. Seu desenvolvimento inicial começou em 1987 pelo Grupo de Trabalho do OSPF da IETF (*Internet Engineering Task Force*). O protocolo é aberto, ou seja, de domínio público, padronizado pelo IETF, independente e não proprietário, podendo ser utilizado gratuitamente por qualquer fabricante.

O OSPF é um protocolo de estado de enlace. Poderíamos pensar em um link como sendo uma interface no roteador. O estado do enlace é uma descrição dessa interface e de sua relação com seus roteadores vizinhos. Uma descrição da interface incluiria, por exemplo, o endereço IP da interface, a máscara, o tipo de rede a que está conectado, os roteadores conectados a essa rede e assim por diante.[10] A coleção de todos esses vínculos formariam um banco de dados.

Toda rota distribuída pelo OSPF possui um endereço de destino e uma máscara de rede. O protocolo OSPF encaminha dados baseados no endereço IP de destino encontrado no cabeçalho do pacote IP. O OSPF detecta qualquer alteração da topologia e calcula novas rotas sem loop após um período de convergência. Todos os roteadores executam o algoritmo do protocolo OSPF em paralelo. Cada roteador constrói uma árvore de menor caminho, com si mesmo como raiz. Esta árvore de menor caminho fornece rotas para cada destino no sistema autônomo.[10]

2.1 Algoritmo do caminho mais curto

O protocolo OSPF usa o algoritmo do caminho mais curto para criar e calcular o caminho mais rápido para todos os destinos conhecidos. O caminho mais curto é calculado com o uso do algoritmo Dijkstra. O algoritmo do OSPF por si só possui um nível muito alto de complexidade, de forma simplificada as várias etapas desse algoritmo são:

1. Após a inicialização ou qualquer alteração nas informações de roteamento, um roteador gera um anúncio de estado de enlace. Este anúncio representa a coleção de todos os dados de estados de enlace nesse roteador.
2. Todos os roteadores trocam essas informações. Cada roteador que recebe uma atualização de estado de enlace deve armazenar uma cópia em seu banco de dados de estados de enlace e, em seguida, propagar a atualização para outros roteadores.
3. Depois que o banco de dados de cada roteador for preenchido, o roteador irá calcular uma árvore de caminho mais curto para todos os destinos. O roteador usa o algoritmo de Dijkstra a fim de calcular a árvore de caminho mais curto. Os destinos, o custo associado e o próximo salto para alcançar esses destinos formam a tabela de roteamento IP.
4. No caso de não ocorrer nenhuma alteração na rede OSPF, como o custo de um enlace ou uma rede que está sendo adicionada ou excluída. Todas as alterações que ocorrem são comunicadas por meio de pacotes de estado de enlace, e o algoritmo de Dijkstra é executado novamente com o intuito de localizar os caminhos mais curtos.

O algoritmo coloca cada roteador na raiz de uma árvore e calcula o caminho mais curto para cada destino com base no custo cumulativo necessário para alcançar esse destino. Cada roteador terá sua própria exibição da topologia, embora todos os roteadores construam uma árvore de caminho mais curta usando o mesmo banco de dados de estado de enlace, no qual cada roteador repete periodicamente o algoritmo acima.[\[10\]](#)

O custo (também chamado de métrica) de uma interface no OSPF é uma indicação da sobrecarga necessária para enviar pacotes através de uma determinada interface. O custo de uma interface é inversamente proporcional à largura de banda dessa interface. Uma largura de banda maior indica um custo mais baixo.

2.2 Interfaces OSPF

Outra idéia importante no OSPF é que interfaces usadas para trocar informações com vizinhos OSPF têm tipos diferentes. Os dois importantes são:

1. Uma interface de broadcast OSPF está conectada a uma rede compartilhada, como Ethernet.
2. Uma interface ponto a ponto OSPF está conectada a um link onde só pode haver um único roteador OSPF em cada extremidade.

A razão para os vários tipos de interface é certificar-se de que todos os roteadores saibam sobre todas as rotas de todos os outros roteadores. Em links ponto-a-ponto, não há nenhum mistério, os dois roteadores sabem que eles são os únicos roteadores OSPF no link e assim eles trocam rotas uns com os outros[12].

2.3 Áreas OSPF

As áreas no OSPF são coleções de roteadores agrupados. Com exceção dos roteadores de borda de área, os roteadores OSPF em uma área não são vizinhos com roteadores em outras áreas. Entre outras razões, as áreas foram usadas para dimensionar grandes redes OSPF.

Quando o poder de processamento dos roteadores era mais fraco do que são hoje, uma regra geral era manter uma área OSPF para não mais de 50 roteadores. Isso manteria o número de cálculos de caminho mais curtos OSPF e atualizações de banco de dados para um valor gerenciável à medida que as interfaces foram para cima e para baixo, as rotas foram aprendidas e retiradas, e assim por diante. Em redes modernas, não é incomum escalar para mil roteadores ou mais.

Embora a escala não seja uma boa razão para implementar várias áreas, as áreas OSPF ainda são úteis como limites administrativos em uma rede. Por exemplo: a recapitulação e a agregação da rota (substituindo várias rotas pequenas com uma rota maior que as cobre) só podem acontecer nos limites da área OSPF. Nem todos os roteadores precisam saber sobre todas as outras rotas disponíveis em uma rede. Usando os conceitos de áreas OSPF, é possível injetar uma rota padrão representando todas as rotas fora da área local.

A área mais importante do OSPF é a área de backbone, também conhecida como área 0. A área de backbone é a área que todas as áreas OSPF devem percorrer para chegar outras áreas. Por exemplo, como foi desenvolvido no projeto, temos a área 0 e a área 1 para cada sistema autônomo. Na área 0 estão todas as interfaces que est ao conectados diretamente com dois sistemas autônomos. A área 1 é representada pelas interfaces que estão conectadas com roteadores pertencentes à mesma rede[12]. Embora os roteadores OSPF dentro de uma área saibam tudo o que há para saber sobre a topologia de rede, as informações de topologia estão ocultas nas bordas da área.

2.4 Vantagens do protocolo OSPF

Após apresentarmos algumas características do protocolo OSPF, apresentaremos algumas vantagens deste protocolo, a fim de demonstrar o porque a maioria dos especialistas em rede preferem os protocolos por estado de conexão.

1. **Convergência Rápida e sem Loop** : Enquanto os protocolos vetor-distância convergem proporcionalmente ao número de nós na rede. Além disso, nos protocolos vetor-distância, a mensagem é proporcional ao número de destinos, se a rede é muito grande, cada mensagem vai ter que ser subdividida em vários pacotes, diminuindo mais ainda a velocidade de convergência. Ainda, nos protocolos de estado do link imediatamente após a transmissão e cálculo, todas as rotas da rede estão sanadas, isto é, não há loops nem contagem ao infinito.
2. **Suporte a várias métricas** : O OSPF V.2 suporta vários tipos de métrica, de forma que a decisão sobre o menor caminho pode ser tomada em relação ao tempo, ao custo por bit ou confiabilidade. Tornando, portanto, a escolha do melhor caminho mais flexível, uma vez que cada meio tem características diferentes. Foi então criada uma extensão indicando o tipo de métrica usada num determinado pacote, para que o nó que o recebeu não use uma outra métrica no seu envio e prejudique o roteamento provocando loops ou atrasos. Esta extensão só é suportada na versão 2 do OSPF.
3. **Caminhos múltiplos** : Nem sempre a melhor rota entre X e Y deve ser a única a ser utilizada, pois isto pode implicar em sua sobrecarga. Análises matemáticas provaram que a divisão do tráfego em duas rotas é muito mais eficiente. Isto, apesar de fazer com que as filas em nós intermediários fiquem desiguais, elas são reduzidas no nós. Além disto, se todos escolhessem uma única rota e ela ficasse indisponível, haveria um grande "re-roteamento" para outro caminho, possivelmente o congestionando. Mas se o tráfego fosse separado em vários caminhos, não haveria muito transtorno caso uma determinada rota ficasse inacessível. O algoritmo que realiza este tipo de análise é bastante complexo, pois, como dificilmente uma fonte e um destino tem duas rotas possíveis exatamente iguais, é feita uma análise se as rotas são suficientemente iguais. Além disto, deve-se decidir a fração do tráfego que deve ser enviado em cada uma delas.

Capítulo 3

Protocolo BGP

O Border Gateway Protocol (Protocolo de Roteamento de Borda - BGP) é um protocolo de roteamento entre sistemas autônomos do tipo vetor de distância (*Distance-Vector*). Criado em 1989 com o intuito de administrar os diálogos entre os roteadores, e de suportar as diferentes infraestruturas já existentes, provendo escalabilidade, flexibilidade e redundância, além de possibilitar a criação de políticas de roteamento que respeitassem as particularidades e os anseios de cada uma das organizações que se conectam com a Internet. BGP é o protocolo usado para troca de informações sobre roteamento da internet e ele é usado por ISP's. Vários clientes compartilham o mesmo AS de um ISP, dessa forma, nos roteadores de borda do AS do ISP é usado o BGP para trocar informações de rotas com a internet para seus clientes. Para que as informações sejam trocadas a partir do protocolo BGP, é feito um acordo, sem levar em consideração o custo da comunicação, onde esse acordo é chamado de peering. O protocolo BGP pode ser usado de duas maneiras: eBGP(External BGP) e iBGP(Internal BGP).

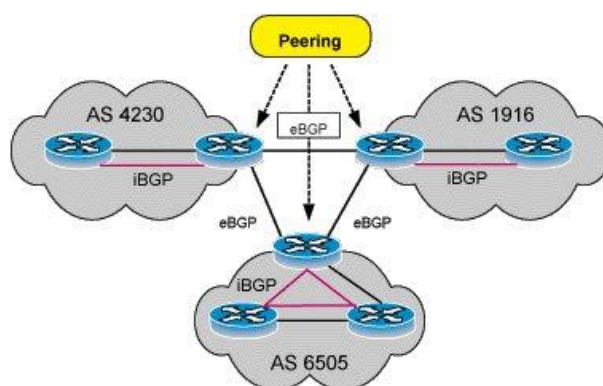


Figura 3.1: Uso do iBGP e do eBGP.[11]

Geralmente um sistema autônomo possui vários roteadores, em que esses roteadores podem trocar informações. Para que isso aconteça, os roteadores devem usar o protocolo BGP interno (iBGP). Já quando dois roteadores de sistemas autônomos diferentes (roteadores de borda), utiliza-se o protocolo BGP externo (eBGP). O BGP usa os mesmos tipos de mensagem nas sessões iBGP e eBGP, mas as regras para quando

enviar cada mensagem e como interpretar cada mensagem diferem ligeiramente. O protocolo BGP possui um atributo chamado *path*, que é o caminho como uma lista de todos os sistemas autônomos que precisam ser percorridos, para alcançar o local onde o prefixo foi anunciado, um meio pelo qual os dispositivos de roteamento BGP evitam loops. Por diferenças entre os protocolos iBGP e o eBGP, eles podem ser considerados dois protocolos separados.

3.1 BGP Interno

O objetivo do protocolo iBGP é fornecer um meio pelo qual os anúncios de rotas eBGP possam ser encaminhados em toda a rede, porém deve haver algum protocolo IGP que permite que os dois vizinhos alcancem um ao outro. Como o tráfego do iBGP não modifica o atributo *path* durante os anúncios, é necessário a prevenção de loops na rede, e para evitar esses loops de roteamento dentro de um sistema autônomo, é essencial que a rede seja de malha completa (todos os roteadores conectados com todos).

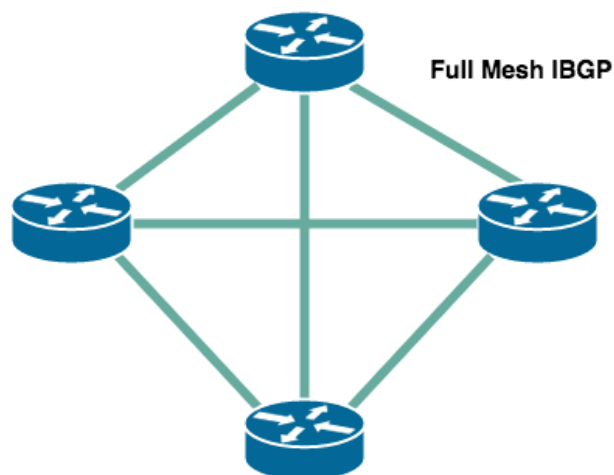


Figura 3.2: Exemplo de um rede de malha completa iBGP.[14]

Nesta topologia, todos os roteadores usados em iBGP devem ser conectados juntos como malha completa. Esse tipo de configuração é complexo e difícil de configurar. A complexidade de criar uma topologia full-mesh é exponencial e pode ser alcançado através da fórmula[13] :

$$n * (n - 1) / 2$$

Sendo n a quantidade de roteadores, e o resultado da expressão o número de peerings necessários para os roteadores se conectarem numa malha completa. Ou seja, em um cenário com 25 roteadores, é necessário a criação de 300 peerings. Outro problema pode ser, quando se torna necessário adicionar um roteador na topologia, precisaria atualizar todos os roteadores na topologia iBGP para este roteador. Um caso para reduzir a complexidade dessa topologia, torna-se necessário utilizar o conceito de refletor de rotas (*router reflector*).

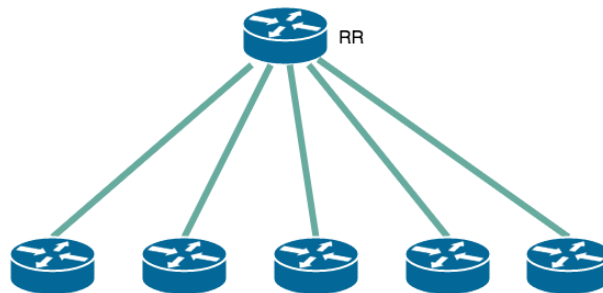


Figura 3.3: Exemplo de um roteador servindo como servidor de rotas para os seus vizinhos.[14]

Para usar o conceito de *router reflector* em um AS, você deve designar um ou mais roteadores que fazem parte de uma malha interior, como um refletor de rotas. Os refletores de rotas têm a capacidade especial do BGP de anunciar as suas rotas aprendidas para outros roteadores internos[7]. Então ao invés de exigir que todos os pontos internos sejam totalmente uma rede de malha completa, a ideia do uso de refletor de rotas exige que apenas os refletores de rota pertençam a rede de malha com todos os pontos internos, onde o refletor de rota e todos os seus pares internos formam um cluster.

A sincronização é um conceito muito importante no protocolo BGP, especialmente quando um AS está fazendo peering entre os AS's vizinhos. Suponha que há dois roteadores do mesmo domínio conectados com um AS em pontos separados, possuindo uma configuração do iBGP para os seus vizinhos internos. Portanto, dois roteadores (A e B) falando eBGP com outro AS e iBGP entre si. Se uma rota para o vizinho do roteador A for anunciada na malha iBGP e, subsequentemente, o roteador B a anunciar para seu par, podemos ter problemas. Pois é provável que o peer de B queira começar imediatamente a enviar tráfego para o peer do outro lado de A. Onde alguns dos roteadores iBGP internos podem não ter conhecimento do caminho para o AS pelo lado de A.

Algumas abordagens diferentes estão disponíveis para lidar com o iBGP e a sincronização. Podemos ativar a opção de sincronização nos nossos roteadores e esperar que o IGP tenha uma rota para o destino antes de ser anunciada aos colegas. Outra opção é simplesmente usar uma malha completa, para que a convergência do iBGP não seja um problema. A alternativa real, se você não ativar a sincronização, é usar a recursão de rota. Uma pesquisa de rota recursiva usa o atributo next-hop do BGP para realmente fazer uma pesquisa de rota diferente. O IGP pode usar a rede de destino em vez do caminho AS para determinar para onde é enviado. Mesmo que o iBGP não tenha convergido, os roteadores ainda saberão como chegar a essa rede, pois ela existirá no roteador de onde foi anunciado, quem saberá o próximo salto.[13]

O protocolo BGP é limitado, em que o valor AS PATH é o único mecanismo externo usado para a seleção de rotas. Dito isso, no entanto, existem alguns atributos do BGP que permitem influenciar o caminho que um pacote leva. O MED, ou MultiExit Discriminator, é usado para indicar um caminho preferido. O MED é essencialmente um peso e o valor mais baixo ganha a preferência. Este é um mecanismo simples para dizer qual ponto de entrada você prefere, se você tiver duas opções para um caminho. O MED é usado para dizer a um colega qual deve ser feito, e ele é passado apenas para seus pares diretos. O atributo LOCAL PREF é usado para informar aos seus pares iBGP a melhor maneira de sair para um AS diferente. Novamente, este é outro mecanismo usado para preferir um caminho igual sobre o outro.

3.2 BGP Externo

O objetivo do protocolo eBGP é anunciar rotas entre sistema autônomo diferentes. O protocolo é implementado nos roteadores de borda, que fornecem essa interconexão entre dois ou mais sistemas autônomos diferentes. Como esse protocolo interage diretamente com outro sistema autônomo, pode se dizer que o protocolo eBGP é o mais importante, pois é a partir dele que se tem o controle do anúncio e importação de rotas para os seus clientes, no qual é definida as políticas e preferências de exportação e importação.

3.3 Critérios na seleção de rotas

Os roteadores BGP aprendem múltiplos caminhos via BGP interno e externo. Eles utilizam somente o melhor caminho e instala na tabela de roteamento IP. O roteador BGP anuncia apenas as rotas que este utiliza (apesar da possibilidade de aprender sobre múltiplos caminhos).

O BGP não tem qualquer métrica simples, as regras para a seleção de uma rota ideal, entre as múltiplas rotas BGP com a mesma preferência, são um pouco mais complexas e são implementadas de acordo com um algoritmo. Começa a primeira regra, se houver mais rotas com o mesmo valor, em seguida, ele usa a segunda regra para escolher entre elas e assim por diante[13]. A seguir temos a lista de como o protocolo BGP, define qual é o melhor caminho.

1. Prefere a rota com o atributo de preferência local mais alto.
2. Prefere a rota com a menor como caminho(path).
3. Preferem a origem do IGP sobre EGP e origem EGP sobre desconhecida.
4. Prefere o menor valor do discriminador de saída múltipla(MED).
5. Prefere rotas recebidas via eBGP sobre uns recebidas via iBGP.

6. Prefere rotas com menor distância interna para um roteador de limite.
7. Prefere a rota com o menor valor de ID de roteador do roteador de anúncio.

O caminho com a preferência de local mais alto é preferencial, ou seja, para as rotas anunciadas a preferência da escolha é a que possuir o valor de preferência mais alto. A segunda regra diz respeito ao caminho percorrido até chegar a rota desejada, uma lista contendo o número de cada sistemas autônomo que pertence a esse caminho, a lista com menos elementos possui a preferência. A terceira regra mostra a preferências de rotas com origem do maior para o menor: IGP, EGP e desconhecida. Já a quarta regra, afirma que a rota que possuir o menor valor MED(discriminador de saída múltipla) detém a preferência, o atributo MED fornece uma maneira dinâmica de influenciar outro sistema autônomo na maneira de alcançar uma determinada rota quando há mais de um ponto de entrada para aquele sistema autônomo. A quinta norma informa que rotas recebidas de outro sistema autônomo tem mais preferência sobre rotas recebidas da mesma rede. Pela sexta lei, avisa que as rotas com menor distância interna para um roteador, tem mais preferência. E o último critério, indica que o menor valor do ID do roteador tem maior preferência.

Para o nosso projeto, dentre essas sete regras, as mais importantes como critérios de escolha são, o valor da preferência local, menor caminho(path) e o valor do atributo MED.

3.4 Importância do uso das políticas BGP

Os provedores de internet(ISP) geralmente desejam controlar a próxima seleção de salto, uma forma de reproduzir os acordos ou relacionamentos que têm com os seus vizinhos. Em base são três relações que o provedores possuem: cliente-servidor, onde um provedor paga outra para transmitir o seu tráfego, ponto a ponto, onde dois provedores concordam em se conectarem diretamente um com o outro(normalmente sem troca de pagamento), onde beneficiária os dois, talvez porque é aproximadamente igual às quantidades de tráfego que fluem entre suas redes, onde dois ISPs configuram uma ligação entre eles que é para ser usado somente no caso em que as principais rotas ficarem indisponíveis devido a falhas.[8]

Intuitivamente os provedores de serviços preferem as rotas aprendidas do seu cliente sobre as rotas aprendidas por outros provedores, quando ambos estão disponíveis. Muitas vezes um provedor vai conseguir obter esse controle através do atributo de preferência local (*LocalPref*)[9], atribuindo um conjunto de valores do *LocalPref*, onde valores de faixa 150-160 podem ser usados para clientes, 100-110 para outros provedores, 90-99 para backups e etc. O atributo do *LocalPref*, pode ser variado dentro de cada intervalo de engenharia de tráfego sem violar as restrições associadas com o relacionamento de negócios.[9] Como um exemplo, um ISP grande, abrangendo a América do Norte e a Europa, pode desejar evitar o encaminhamento de tráfego gerado pelos seus clientes através de um link caro transatlântico. Isso pode ser feito, ao configurar

seus roteadores europeus com um LocalPref mais elevado, para as rotas aprendidas com ISPs europeus e dando a seus roteadores norte-americanos um LocalPref inferior para essas rotas.[8]

As rotas aprendidas pelos provedores de serviços geralmente não são exportados para outros provedores, porque não há nenhum incentivo econômico para um ISP encaminhar o tráfego que recebe de um provedor para outro. Isso pode ser feito por propagandas de marcação, através do atributo community(comunidade), significando o relacionamento de negócio da sessão e a filtragem de rotas com determinados atributos do community, quando são exportados sua rotas para os seus sistemas autônomos vizinhos.[8] Os valores do atributo community são tratadas como valores de 32 bits.[8]

Projeto e a sua implementação

O projeto consiste na implementação de um ambiente, onde podemos estudar e aplicar os conceitos do protocolo BGP, tal como na tomada de decisão da melhor rota possível, políticas de importação e exportação, entre outros relacionamentos. Para que isso seja feito, simulamos quatro sistemas autônomos, onde as conexões entre eles podem ser vistas na Figura 1.1. Para o cálculo do custo interno de cada sistema autônomo é utilizado o protocolo OSPF e para a comunicação entre sistemas autônomos, é utilizado o protocolo BGP.

Cada sistema autônomo possui seus próprios roteadores, o sistema autônomo da RNP, contém os roteadores internos, baseado na estrutura real da própria RNP, como visto na Figura 4.1.

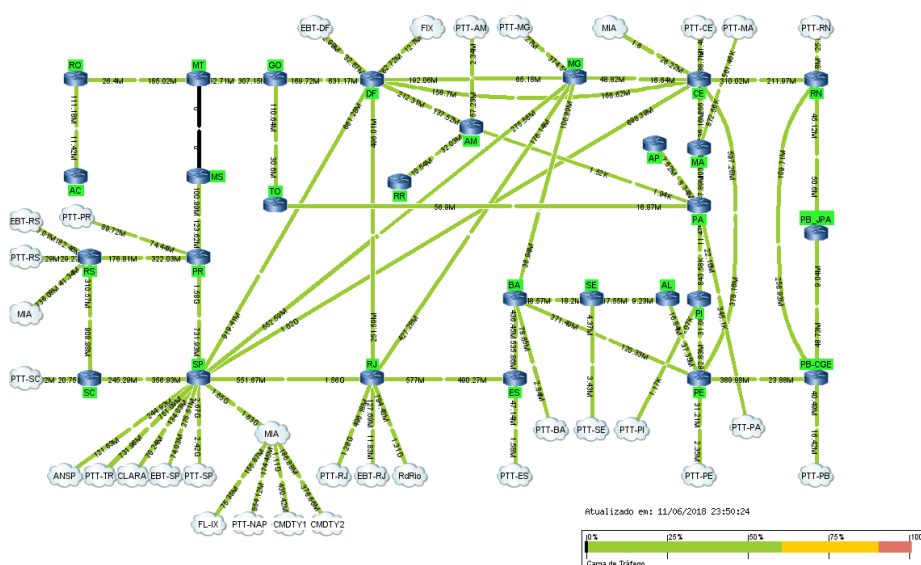


Figura 4.1: Roteadores interno do sistema autônomo da RNP

Como não temos uma rede de malha completa, é necessário a utilização do conceito de refletor de rotas (*route reflector*), onde os roteadores *SP*, *DF*, *MG* e *CE* são refletores de rotas dos seus roteadores vizinhos, e esses vizinhos sendo refletores de rotas dos próximos, assim em diante, até abranger todos os roteadores. No nosso projeto, temos que os roteadores *SP*, *DF*, *RS*, *CE*, fazem peering com o AS da Embratel e os roteadores *SP*, *RS*, *CE*, fazem peering com o AS da I2. Esses roteadores que fazem peering com os outros sistemas autônomos exportam rotas pertencentes a cada roteador da rede RNP, e como temos várias saídas para cada um dos ASes, para que seja escolhida o caminho, torna-se necessário a utilização do atributo MED, tal que o valor MED exportado de cada rota, é o valor do custo do roteador que exporta o prefixo para o AS vizinho, até o roteador de origem do mesmo.

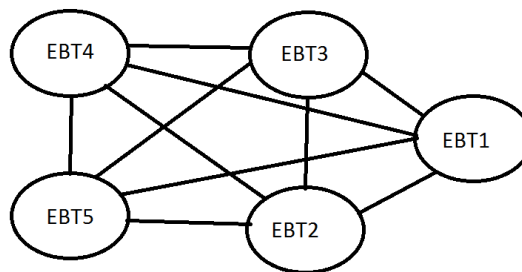


Figura 4.2: Roteadores internos simulados que fazem parte do sistema autônomo da Embratel

Já o sistema autônomo da Embratel consiste em uma rede de malha completa, contendo cinco roteadores, em que o roteador *EBT1* faz peering com um roteador da rede I2 e com o roteador *SP* da RNP, o roteador *EBT2* faz peering com o roteador *CE* da RNP, o roteador *EBT3* faz peering com o roteador do AS BF (Backbone Final), o roteador *EBT4* faz peering com o roteador *RS* da RNP, e por fim o roteador *EBT5* faz peering com o roteador *DF* da RNP. A respeito do peering entre o AS da Embratel com o AS da RNP, temos uma relação política de exportação, no qual as rotas exportadas do AS da Embratel para a RNP, não podem ser exportadas para nenhum outro AS.

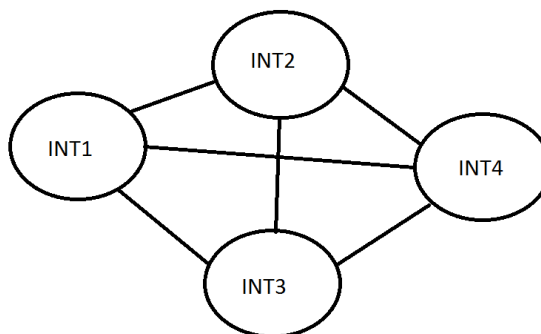


Figura 4.3: Roteadores internos simulados que fazem parte do sistema autônomo I2

O sistema autônomo I2 (Internet Comercial 2), representa uma rede de malha completa, contendo quatro roteadores. O roteador *INT1* faz peering com o AS da Embratel e com o roteador *SP* da RNP, o roteador *INT2* faz peering com o roteador do AS BF(backbone final), já o roteador *INT3* faz peering com o roteador *CE* da RNP, e por fim o roteador *INT4*, que faz peering com *RS* da RNP. A respeito do peering entre o AS da I2 com o AS da RNP, temos uma relação política de exportação de rotas, em que as rotas exportadas do AS da I2 para a RNP, não podem ser exportadas para nenhum outro AS.

E por fim o AS BF (Backbone final), representado apenas por um roteador que faz peering com os ASes da Embratel e I2, o objetivo desse sistema autônomo é visualizar a tomada de decisão do melhor caminho para as rotas originadas do AS da RNP, visto que temos dois caminhos, através do AS da Embratel ou do AS da I2. A decisão da melhor rota é feita a partir do primeiro critério, que se refere ao maior valor do LocalPref tem a preferência mais relevante.

A implementação desses quatro sistemas autônomos foi feita a partir de duas partes, o plano de dados e o plano de controle. O kernel do Linux foi encarregado na parte do plano de dados, ou seja, a criação dos roteadores, interfaces, links de enlace e atribuições de endereços criados utilizando o Linux Network Namespace. O software BIRD foi responsável pelo plano de controle, ou seja, o BIRD é responsável pela definição dos protocolos de roteamento que foram usados, manuseando cada protocolo para que cada roteador realize o seu trabalho de forma correta, e o relacionamento entre os roteadores criados.

O nome "BIRD" é na verdade um acrônimo para "BIRD Internet Routing Daemon". Pela definição encontrada no site dos desenvolvedores, o BIRD é um programa que funciona como um roteador dinâmico em uma rede do tipo Internet (isto é, em uma rede que executa o protocolo IPv4 ou IPv6). Os roteadores são dispositivos que encaminham pacotes entre redes interconectadas para permitir que hosts não conectados diretamente à mesma rede local se comuniquem uns com os outros. Eles também se comunicam com os outros roteadores na Internet para descobrir a topologia da rede que permite encontrar regras ótimas (em termos de algumas métricas) para o encaminhamento de pacotes (que são chamados de tabelas de roteamento) e se adaptar às condições de mudança, como interrupções de links de rede, construção de novas conexões e assim por diante. A maioria desses roteadores são dispositivos dedicados e dispendiosos que executam firmware obscuro, difícil de configurar e que não se abrem a nenhuma alteração.[3]

Introduzido em 2002, na versão 2.4.19 do kernel Linux, Namespace é um recurso que permite criar e lidar com diversos contextos em um mesmo sistema, vendo propriedades globais diferentes e isoladas em cada contexto, ou seja, permite criar diferentes ambientes independentes que são executados no sistema base. Para a criação desses ambientes são utilizados alguns recursos do Namespace[5], tal como: mount namespaces (*mnt*), process id namespaces (*pid*), unix timesharing system namespace (*uts*), network namespace (*net*), inter-process communication namespace (*ipc*), user namespace (*usr*). O mount namespaces é responsável por criar um ambiente isolado para

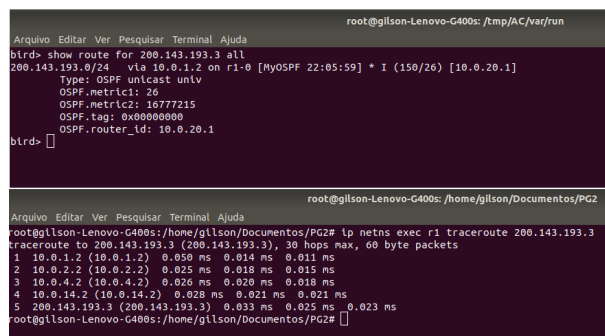
os dispositivos que podem ser montados pelo sistema[4]. O pid é um indetificador de cada processo do kernel[2]. O uts namespace é um recurso usado para isolar dois elementos específicos do sistemas que se relacionam com uma chamada de sistema[5]. Network namespaces permite criar um ambiente de rede isolado do ambiente físico, nesse ambiente existirão interfaces de rede física, que possuem endereços físicos e lógicos diferentes[1]. O ipc namespace tem o intuito de isolar processos de comunicação[5]. O user namespace é responsável pelo isolamento dos identificadores e atributos relacionados à segurança.[6]

Capítulo 5

Avaliação da Implementação

Para apresentar a avaliação da implementação, vamos mostrar nesta seção, as tomadas de decisão realizadas pelos sistemas autônomos implementados, tal como a comunicação interna, utilização do route reflector, ação do protocolo iBGP, importância do atributo MED no protocolo eBGP, implementações de políticas a partir do atributo community, e a atuação do valor do LocalPref. Para demonstrar o comportamento desses aspectos, foi utilizado os recursos *traceroute* (Rastreia a rota de um pacote através de uma rede de computadores que utiliza os protocolos IP e o ICMP), *fping* (Encontra as máquinas conectadas e ligadas em uma rede) e as tabelas de roteamento dos roteadores criados.

Como já foi dito nos capítulos anteriores, o protocolo usado para a comunicação interna foi o OSPF, em que a função dele é calcular os saltos de um roteador para outro, e escolher o de menor caminho. Na implementação, cada salto de um roteador para outro tem um custo de valor 5, e o custo do salto para o endereço privado de cada roteador é de 1. Como exemplo, o caminho do roteador AC até o roteador RJ, temos o custo de 26, como visto na Figura 5.1 através do atributo "OSPF.metric", pois temos 5 saltos entre esses roteador e o custo de mais um para acessar o endereço privado do roteador RJ, pode ser visto na parte superior da Figura 5.1.



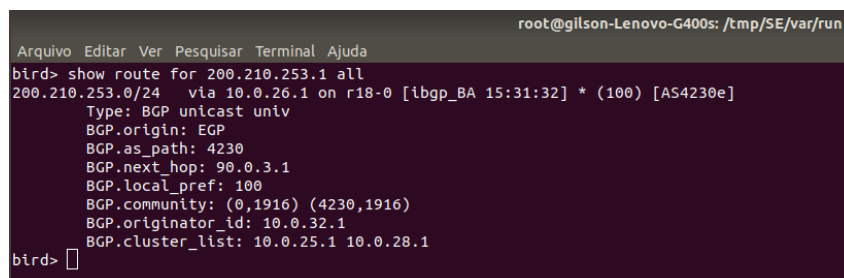
```
root@gilson-Lenovo-G400s: /tmp/AC/var/run
bird> show route for 200.143.193.3 all
200.143.193.0/24 via 10.0.14.2 on r1-0 [MyOSPF 22:05:59] * I (150/26) [10.0.20.1]
  Type: OSPF unicast unlv
  OSPF.metric1: 26
  OSPF.metric2: 16777215
  OSPF.tag: 0x00000000
  OSPF.router_id: 10.0.20.1
bird>

root@gilson-Lenovo-G400s: /home/gilson/Documentos/PG2
root@gilson-Lenovo-G400s: /home/gilson/Documentos/PG2# ip netns exec r1 traceroute 200.143.193.3
traceroute to 200.143.193.3 (200.143.193.3), 30 hops max, 60 byte packets
 1 10.0.1.2 (10.0.1.2) 0.050 ms 0.014 ms 0.011 ms
 2 10.0.2.2 (10.0.2.2) 0.025 ms 0.018 ms 0.015 ms
 3 10.0.4.2 (10.0.4.2) 0.026 ms 0.020 ms 0.018 ms
 4 10.0.14.2 (10.0.14.2) 0.028 ms 0.021 ms 0.021 ms
 5 200.143.193.3 (200.143.193.3) 0.033 ms 0.025 ms 0.023 ms
root@gilson-Lenovo-G400s: /home/gilson/Documentos/PG2#
```

Figura 5.1: A parte superior da imagem, mostra as informações da tabela de roteamento do roteador AC, para o prefixo que é originado pelo roteador RJ. A parte inferior da imagem, mostra os saltos, a partir do roteador AC até o RJ.

A Figura 5.1 é dividida por duas imagens, no qual a primeira, podemos ver o custo do roteador AC até o roteador RJ, em que o prefixo 200.143.193.0/24 pertence ao roteador RJ. Como visto na imagem, esse custo é 26, podemos visualizar isso, a partir do campo *OSPF.metric1*. Já a imagem de baixo, representa os saltos que o roteador AC deve realizar para se comunicar com o roteador RJ. Os endereços 10.0.1.2, 10.0.2.2, 10.0.4.2, 10.0.14.2, 200.143.193.0/24, representam os links de enlace dos roteadores RO, MT, GO, DF e RJ, respectivamente.

Conforme a seção anterior, a utilização do recurso route reflector foi a partir dos roteadores SP, MG, DF, CE, para os seus vizinho e assim em diante, formando assim uma lista de refletores de rotas até o seu destino. Temos um prefixo exportado pelo AS da Embratel, e transmitido entre os roteadores internos pelo protocolo iBGP.



```
root@gilson-Lenovo-G400s: /tmp/SE/var/run
Arquivo Editar Ver Pesquisar Terminal Ajuda
bird> show route for 200.210.253.1 all
200.210.253.0/24 via 10.0.26.1 on r18-0 [ibgp_BA 15:31:32] * (100) [AS4230e]
Type: BGP unicast univ
BGP.origin: EGP
BGP.as_path: 4230
BGP.next_hop: 90.0.3.1
BGP.local_pref: 100
BGP.community: (0,1916) (4230,1916)
BGP.originator_id: 10.0.32.1
BGP.cluster_list: 10.0.25.1 10.0.28.1
bird> □
```

Figura 5.2: A imagem de cima, mostra as informações da tabela de roteamento do roteador SE, para o prefixo que é originado pelo roteador EBT1 da AS da Embratel.

Com a Figura 5.2, através do atributo "BGP.cluster", podemos ver a lista de identificadores dos roteadores que funcionam como refletores de rota. Até o roteador vizinho do ID pertencente ao atributo "BGP.originator". Os prefixos 10.0.25.1, 10.0.28.1 são os números de ID dos roteadores BA, MG, respectivamente. Ou seja, o roteador MG reflete as rotas para o roteador BA, e o roteador BA reflete as rotas para o roteador SE, assim sendo possível uma comunicação do protocolo iBGP de forma correta com outros roteadores da rede. O ID de número 10.0.32.1 pertence ao roteador CE, que faz peering com o roteador EBT2 do AS da Embratel. Logo, para comunicação do roteador "SE" para o roteador EBT1, passa pelo roteador de borda CE.

A utilização do atributo MED torna-se necessário, pois nos sistemas autônomos da Embratel e da I2, temos múltiplas saídas para a rede da RNP. Como foi visto na seção 3, o quarto critério de escolha na melhor rota é o valor MED, tal que o ponto de saída com o valor do MED mais baixo é preferido. Para a atribuição desse valor, foi usado o custo calculado pelo protocolo OSPF, como é feito na comunicação interna, e esse custo de cada rota é exportado para outros ASes como o valor do atributo MED.

Para o roteador que importa essas rotas com os atributos MED exportam para os seus vizinhos da mesma rede esses prefixos, no qual o roteador escolhe a rota que possuir o menor valor MED. Como podemos visualizar na Figura 5.3.

```

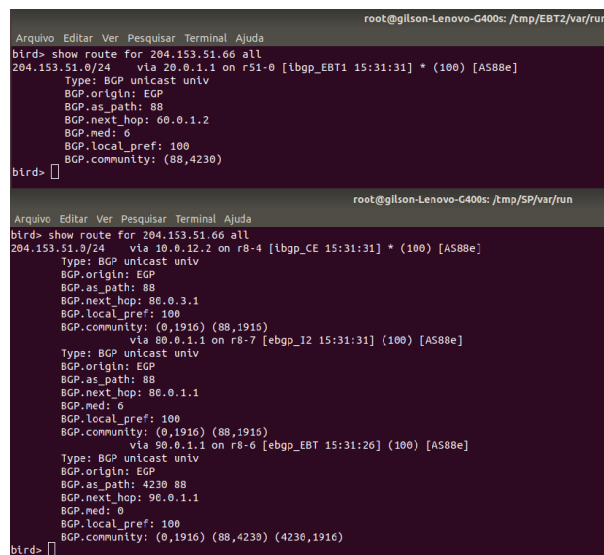
bird> show route for 200.133.192.80 all
200.133.192.0/24 via 30.0.2.1 on r72-0 [ibgp_I2_1 15:31:31] * (100) [AS1916e]
  Type: BGP unicast univ
  BGP.origin: EGP
  BGP.as_path: 1916
  BGP.next_hop: 80.0.1.2
  BGP.local_pref: 100
  BGP.community: (1916,88)
    via 80.0.3.2 on r72-8 [ebgp_CE 15:31:31] (100) [AS1916e]
  Type: BGP unicast univ
  BGP.origin: EGP
  BGP.as_path: 1916
  BGP.next_hop: 80.0.3.2
  BGP.med: 6
  BGP.local_pref: 100
  BGP.community: (1916,88)
bird>
root@glilson-Lenovo-G400s:/home/gilson/Documentos/backup/PG2# ip netns exec r72 t
traceroute -s 204.153.51.65 200.133.192.80
traceroute to 200.133.192.80 (200.133.192.80), 30 hops max, 60 byte packets
 1 30.0.2.1 (30.0.2.1) 0.559 ms 0.484 ms 0.446 ms
 2 200.133.192.80 (200.133.192.80) 0.421 ms 0.375 ms 0.345 ms
root@glilson-Lenovo-G400s:/home/gilson/Documentos/backup/PG2#

```

Figura 5.3: A imagem de cima mostra as informações da tabela de roteamento do roteador *INT3*, para o prefixo que é originado pelo roteador *SP* do AS da RNP, e a rota de um pacote a partir do roteador *INT3* até *SP*

Na imagem acima, podemos visualizar como o atributo MED fez a diferença, na escolha do melhor caminho do roteador "INT3" para o prefixo *200.133.192.80/24*, pertencente ao roteador *SP* do sistema autônomo da RNP. Temos duas rotas, uma com o valor MED nulo, e a outro com o valor do atributo MED igual a 6. Isso porque a rota com o valor do MED nulo foi exportada pelo roteador *INT1*(30.0.2.1) devido o protocolo "ibgp_I2_1", onde o *INT1* realiza o peering com o roteador *SP*, logo como faz conexão direta, o custo do valor MED é nulo. Já o roteador *CE* exporta o valor MED com o mesmo custo até o roteador *SP*. Por fim o roteador *INT3* escolhe com a prioridade mais alta, a rota que possui o valor MED nulo.

Conforme já foi dito na seção 3.4, a funcionalidade do atributo community consiste que as rotas exportadas do AS da EBT para o AS da RNP não podem ser exportadas para num sistema autônomo, o mesmo vale com o AS da I2, onde as rotas exportadas para a RNP, também não podem ser exportadas para outros AS'es. Para cada rota que pertence a essa política, é marcada com o valor 0 e com o número do AS a qual vai importar, e a partir de uma tomada de decisão, essa rota é exportada ou não. A Figura 5.4 mostra a visualização da marcação das rotas.



```
root@gilson-Lenovo-C400s: /tmp/EBT2/var/run
Arquivo Editar Ver Pesquisar Terminal Ajuda
bird> show route for 204.153.51.66 all
204.153.51.0/24 via 20.0.1.1 on r51-0 [tbgp_EBT1 15:31:31] * (100) [AS88e]
Type: BGP unicast univ
BGP.origin: EGP
BGP.as_path: 88
BGP.next_hop: 00.0.1.2
BGP.med: 0
BGP.local_pref: 100
BGP.community: (88,4230)
bird>

root@gilson-Lenovo-C400s: /tmp/SP/var/run
Arquivo Editar Ver Pesquisar Terminal Ajuda
bird> show route for 204.153.51.66 all
204.153.51.0/24 via 10.0.12.2 on r8-4 [tbgp_CE 15:31:31] * (100) [AS88e]
Type: BGP unicast univ
BGP.origin: EGP
BGP.as_path: 88
BGP.next_hop: 88.0.3.1
BGP.local_pref: 100
BGP.community: (0,1916) (88,1916)
via 88.0.1.1 on r8-7 [ebgp_I2 15:31:31] (100) [AS88e]
Type: BGP unicast univ
BGP.origin: EGP
BGP.as_path: 88
BGP.next_hop: 88.0.1.1
BGP.med: 0
BGP.local_pref: 100
BGP.community: (0,1916) (88,1916)
via 98.0.1.1 on r8-6 [ebgp_EBT 15:31:26] (100) [AS88e]
Type: BGP unicast univ
BGP.origin: EGP
BGP.as_path: 4230 88
BGP.next_hop: 98.0.1.1
BGP.med: 0
BGP.local_pref: 100
BGP.community: (0,1916) (88,4230) (4230,1916)
bird>
```

Figura 5.4: A imagem de cima, mostram duas imagens, a de cima mostra as informações da tabela de roteamento do roteador *EBT2*, para o prefixo *204.153.51.66* pertencente ao roteador *INT3*. a imagem de baixo mostra as informações da tabela de roteamento do roteador *SP*, para o mesmo prefixo da imagem de cima.

Com a Figura 5.4 podemos visualizar na parte inferior da imagem, que as rotas do AS I2, são marcadas com dois atributos community, em que um é marcado com (0, 1916), significa que o AS de número 1916 (RNP) não pode exportar essa rota nenhum outro sistema autônomo, e o segundo atributo marcado (88,1916), significa que essa rota pode ser importada do AS 88 (I2) para o AS 1916 (RNP). Já na imagem de cima mostra que essa rota que foi marcada com esse atributo community não foi exportada do AS da RNP para o AS da EBT.

E por fim, apresentaremos o primeiro critério de escolha do melhor caminho. Para isso selecionamos todas as rotas que o BF importa do AS da Embratel, com um valor de LocalPref maior que as rotas importadas do AS da I2. Na Figura 5.5, podemos ver o valor do LocalPref decidindo a escolha do caminho para tal prefixo.

```

bird> show route for 200.129.156.1 all
200.129.156.0/24 via 99.99.1.1 on r99-0 [ebgp_EBT 02:07:10] * (100) [AS1916e]
    Type: BGP unicast univ
    BGP.origin: EGP
    BGP.as_path: 4230 1916
    BGP.next_hop: 99.99.1.1
    BGP.med: 0
    BGP.local_pref: 150
    BGP.community: (1916,4230) (4230,3031)
    via 99.99.2.1 on r99-1 [ebgp_I2 02:07:14] (100) [AS1916e]
    Type: BGP unicast univ
    BGP.origin: EGP
    BGP.as_path: 88 1916
    BGP.next_hop: 99.99.2.1
    BGP.med: 0
    BGP.local_pref: 100
    BGP.community: (88,3031) (1916,88)
bird>
root@gilson-Lenovo-G400s:/home/gilson/Documentos/backup/PG2# ip netns exec r99 t
traceroute -s 199.99.99.99 200.129.156.1
traceroute to 200.129.156.1 (200.129.156.1), 30 hops max, 60 byte packets
 1 99.99.1.1 (99.99.1.1) 0.307 ms 0.261 ms 0.248 ms
 2 20.0.11.2 (20.0.11.2) 0.238 ms 0.220 ms 0.207 ms
 3 90.0.4.2 (90.0.4.2) 0.196 ms 0.178 ms 0.164 ms
 4 200.129.156.1 (200.129.156.1) 0.153 ms 0.132 ms 0.117 ms
root@gilson-Lenovo-G400s:/home/gilson/Documentos/backup/PG2#

```

Figura 5.5: A imagem mostra a tabela de roteamento e a rota de um pacote saído do roteador da rede BF (representado com o nome *FN*), para o prefixo *200.129.156.1*, pertencente ao roteador AM do AS da RNP.

A partir da Figura 5.5, podemos verificar como o atributo LocalPref influencia na decisão do caminho para um prefixo, onde temos o valor desse atributo vindo do AS da Embratel igual a 150, já o valor de preferência local vindo do AS da I2 é igual a 100. Portanto para o roteador do BF se comunicar com qualquer roteador da RNP, essa comunicação passará preferencialmente pela rede da Embratel, devido ao valor de preferência local ser maior, sendo mostrado na parte inferior da figura, no qual os saltos de um pacote, a partir do BF até AM segue a ordem dos roteadores 99.99.1.1(BF), 20.0.11.2(EBT5), 90.0.4.2(DF), 200.129.156.1(AM).

Capítulo 6

Conclusão

Este projeto abordou aspectos relevantes do protocolo BGP e a sua utilidade entre os roteadores de borda, com o intuito de interligar as redes de sistemas autônomos diferentes. Com o objetivo de estudar o entendimento de como é o funcionamento dessa comunicação, apresentou-se nesse trabalho, aspectos específicos do funcionamento do protocolo, exibindo métodos de exportação e importação de prefixos, configuração das tabelas de roteamento.

Da mesma forma, esse projeto apresentou brevemente, os passos básicos das configurações necessárias para obter o objetivo geral deste trabalho, que possibilita a criação e a configuração de uma topologia utilizando o protocolo BGP nos roteadores que fazem essa comunicação com outros ASes. Onde pode ser aplicado em um ambiente real, podendo interligar cidades, países, continentes, regiões e etc. Melhorando a qualidade de conexão e transferência de dados entre essas redes.

Com os conhecimentos apresentados, o objetivo desse projeto foi alcançado, pois é possível demonstrar as necessidades da implementação e investimentos necessários para a uma configuração baseada em negócios, bom como as aplicações de políticas necessários para a configuração e manutenção de um ambiente de roteadores, possibilitando uma melhor saída de dados para as aplicações da internet.

Referências Bibliográficas

- [1] *IP-NETNS(8) Linux*, JANUARY 2013. 4
- [2] *pid_namespaces(7) Linux Programmer's Manual*, NOVEMBER 2017. 4
- [3] BIRD: Internet routing daemon. <http://bird.network.cz>, 2018. 4
- [4] *mount_namespaces(7) Linux Programmer's Manual*, APRIL 2018. 4
- [5] *namespace(7) Linux Programmer's Manual*, FEBRUARY 2018. 4
- [6] *user_namespaces(7) Linux Programmer's Manual*, FEBRUARY 2018. 4
- [7] C. Bates and R. Chandra. BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP). RFC 4271, RFC Editor, July 2006. 3.1
- [8] M. Caesar and J. Rexford. Bgp routing policies in isp networks. *Netwrk. Mag. of Global Internetwkg.*, 19(6):5–11, Nov. 2005. 3.4
- [9] T. P. T. L. Chandra, R. BGP Communities Attribute. RFC 1997, RFC Editor, August 1996. 3.4
- [10] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach (6th Edition)*. Pearson, 6th edition, 2012. 2, 2.1
- [11] F. Madeira. BGP - Border Gateway Protocol. <http://www.madeira.eng.br/wiki/index.php?page=BGP+-+Border+Gateway+Protocol>, 2008. 3.1
- [12] J. Moy. OSPF Version 2. RFC 2328, RFC Editor, April 1998. 2.2, 2.3
- [13] E. L. T. E. Rekhter, Y. and E. S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, RFC Editor, January 2006. 3.1, 3.1, 3.3
- [14] D. Vieceli. Escalando o BGP com o uso de Router Reflectors. <https://supportforums.cisco.com/t5/routing-switching-documentos/escalando-o-bgp-com-o-uso-de-router-reflectors/ta-p/3164370>, 2016. [Online; accessed 04-July-2018]. 3.2, 3.3