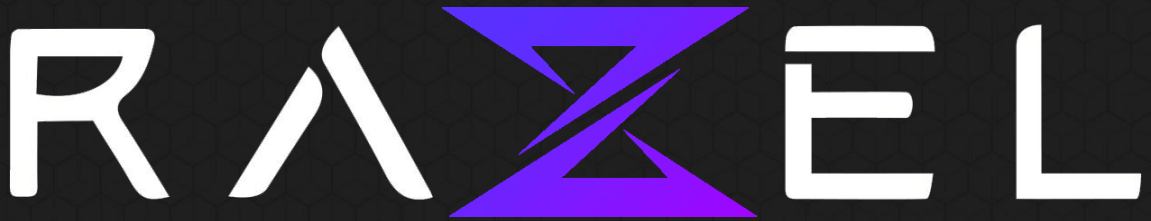
Innovational thinking, we get IT working



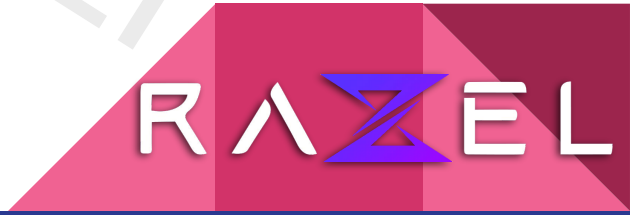
BGP 4.0 (Juniper Networks)

Por Caio Fratelli

Há 20 anos no mercado de TI, Caio conseguiu acompanhar o surgimento e a ascensão da Internet no Brasil.
Cursou Ciências da Computação na faculdade UNIRP.
Suas principais áreas de atuação foram: Infraestrutura/Servidores, Redes e Desenvolvimento.
Ao longo desses anos, teve a oportunidade de realizar projetos para Provedores (ISPs) e Operadoras (ITPs) em todo o país, tendo com destaque seu trabalho junto à Sinal Br, uma operadora regional referência no Noroeste Paulista, ao qual, trabalhando como Consultor e Engenheiro de Redes pôde ser responsável pela implantação da malha MPLS e pela alta disponibilidade do cenário.
Hoje, além de Fundador e CEO da RAZEL, Caio junto a sua equipe ministra treinamentos em Redes, realiza serviços de consultoria para provedores em todo território nacional, assessoria e desenvolvimento de softwares voltado ao mercado de Redes.

Caio Fratelli

Disclaimer



Índice deste treinamento

- O que são Sistemas Autônomos
- Como funciona a internet
- O que é e para que serve o protocolo BGP
- Tipos de BGP
- Estados do protocolo BGP
- Atributos BGP
- Communities BGP
- Configurações de sessões iBGP e eBGP
- Route-Filters
- Laboratório

O que são Sistemas Autônomos?

A internet nada mais é do que a definição da malha que é composta por diversos sistemas autônomos.

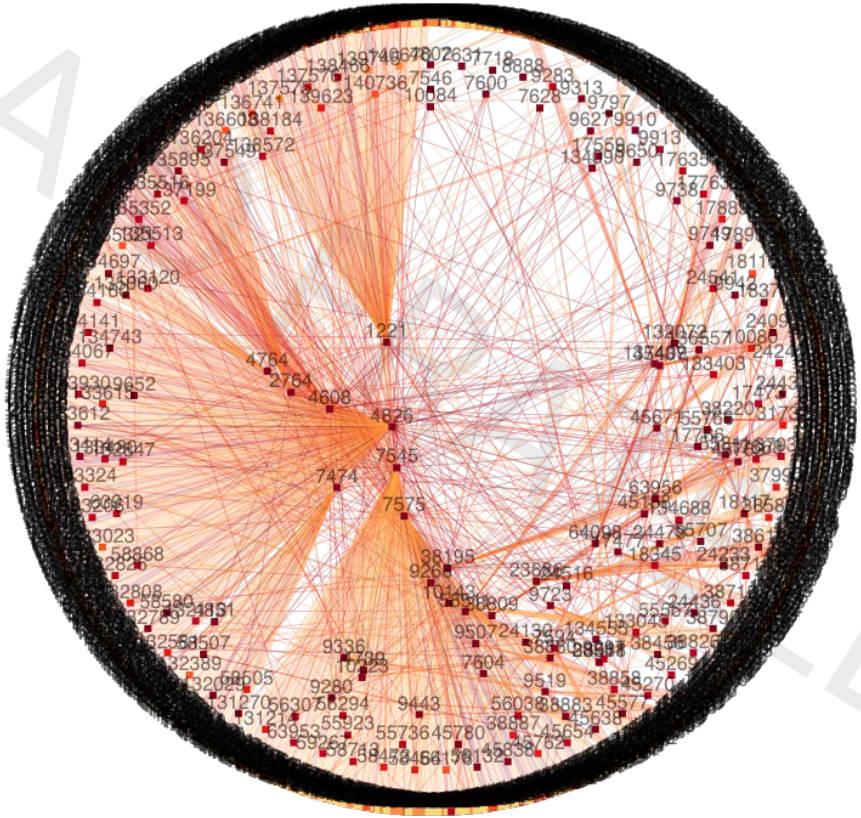
Um sistema autônomo é uma entidade que possui seus próprios recursos de numeração (IPs), seu número de identificação único (ASN - Autonomous System Number) e que por possuir autoridade sobre seus prefixos, pode os anunciar para qualquer outra operadora, ou seja, qualquer outro sistema autônomo.

Quando um sistema autônomo compra conectividade de outro sistema autônomo, chamamos de compra de **trânsito**.

Como funciona a internet?

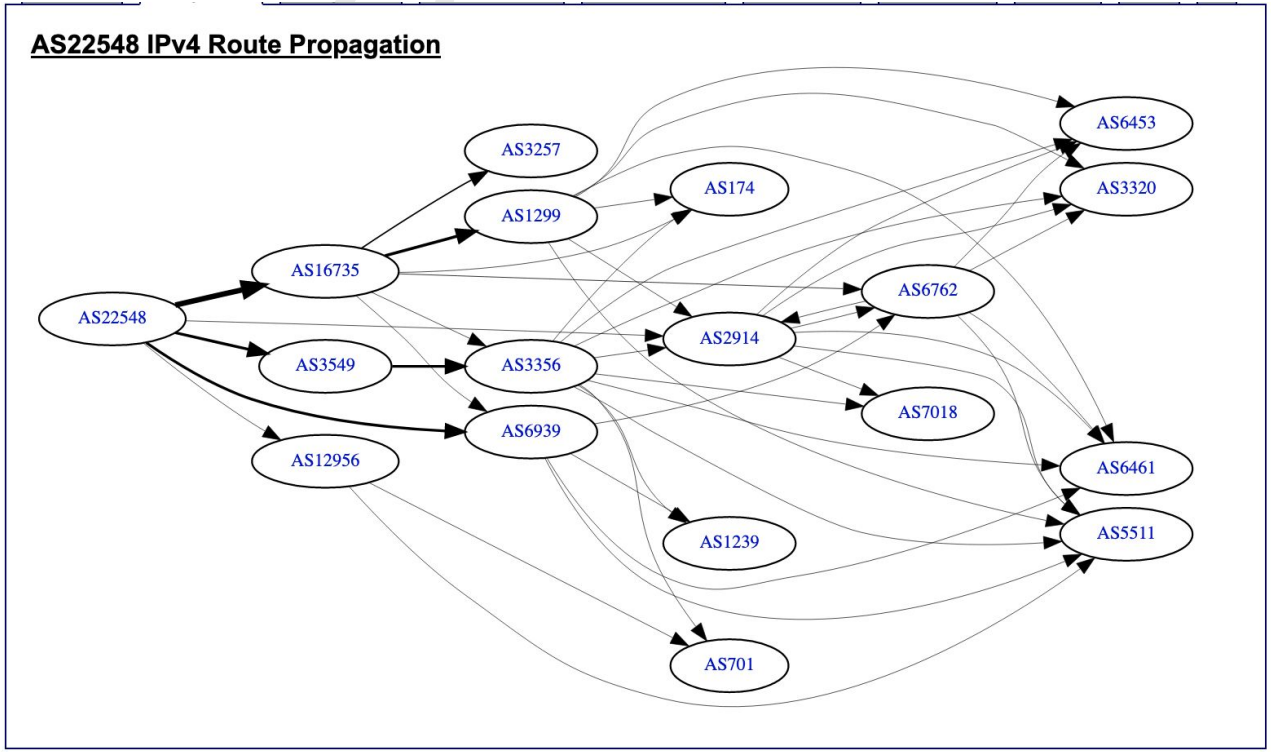
RAZEL

Como funciona a internet?



Fonte: APNIC REx

Como funciona a internet?



Fonte: Hurricane Electric

O que é e para quê serve o protocolo BGP

O BGP ou Border Gateway Protocol versão 4 é um protocolo utilizado por Sistemas Autônomos no roteador que trabalha no limite de suas redes (borda) seja para conectividade externa ou interna para divulgação e recebimento de rotas de um determinado sistema autônomo e seleção do melhor caminho de acordo com a diretriz do protocolo (path vector).

"Tipos" de BGP

De uma maneira superficial, há dois "tipos" básicos de BGP:

iBGP e eBGP (os quais serão explanados nos slides posteriores)

Dentro desses dois tipos há também as extensões como o MPBGP (significa MultiProtocol BGP) ou MBGP (há divergências em relação à nomenclatura em diferentes literaturas), nesta modalidade, o protocolo BGP é apenas um meio de transporte e serve para carregar informação de outros protocolos (NLRIs) que não tem nenhuma relação com Sistemas Autônomos..
Todavia MPBGP foge do espectro deste curso.

"Tipos" de BGP

iBGP – internal BGP ou iBGP é a modalidade que trata das rotas internas, no caso de um mesmo ASN. (Exemplo: para divulgação das rotas de uma mesma empresa entre matriz e filial ou entre múltiplos roteadores da organização)

Os principais diferenciais do iBGP são:

a) Multihop por padrão
b) Aceita rotas vindas do próprio AS em questão
c) Forma vizinhança somente com ASNs de mesmo número.
d) Tem necessidade de topologia Full Mesh.

Tipos de BGP

eBGP – external BGP ou eBGP é a modalidade que trata da troca de rotas entre Sistemas Autônomos diferentes ao redor do mundo onde entre eles é fornecido trânsito.

As principais características do eBGP são:

a) Forma vizinhança apenas com ASNs diferentes do local
b) Não é multihop por padrão
c) Não aceita rotas que possuam o seu próprio ASN no AS_PATH
d) Não tem necessidade de Full Mesh.

Estados do protocolo BGP

No estabelecimento da sessão BGP, o protocolo passa por seis estados diferentes, são eles:

a) **Idle**
b) **Connect**
c) **Active**
d) **OpenSent**
e) **OpenConfirmed**
f) **Established**

Estados do protocolo BGP

Quando a sessão se encontra no estado **Idle**, ela obedece aos seguintes critérios:

a) Recusa todas sessões BGP entrantes.
b) Inicializa os processos para evolução para os próximos estados.
c) Inicia uma conexão TCP com seu peer BGP configurado
d) Escuta por uma conexão TCP do seu peer
e) Muda de estado para **Connect**

Nota: *Se ocorrer um erro em qualquer um dos seis estados, a sessão BGP é finalizada imediatamente e volta ao estado* ***Idle.*** *Algumas razões que podem causar o estado da sessão para ficar preso em* ***Idle*** *são:*

a) A porta 179 TCP está filtrada
b) Uma porta TCP randômica acima de 1023 está filtrada
c) Endereços de Neighbor estão configurados incorretamente em algum dos routers
d) Número do AS está incorreto em um dos routers
e) Requerimento para multihop é requerido porém não foi habilitado.
f) Foi especificado um local address ou local-as incorreto

Estados do protocolo BGP

Quando a sessão se encontra no estado **Connect**, obedece aos seguintes critérios:

a) Espera por uma negociação TCP positiva do neighbor
b) O BGP não perde muito tempo nesse estado se a sessão TCP tiver sido estabelecida corretamente.
c) Envia a mensagem **Open** para o neighbor e muda de estado para **OpenSent**.
d) Se ocorrer um erro, o BGP irá alternar para o estado **Active**.

Os mesmos erros citados no slide anterior podem ser causa para uma sessão travada no modo **Active**.

Estados do protocolo BGP

Quando a sessão se encontra no estado **Active**, obedece aos seguintes critérios:

a) Se o roteador não conseguir estabelecer uma sessão TCP corretamente, ele irá alternar de estado para **Active**.
b) O processo de seis estados BGP chamado FSM tentará reiniciar outra sessão TCP com o neighbor e se obtiver sucesso, irá enviar uma mensagem **Open** para o neighbor.
c) Se falhar novamente, o estado será alterado para **Idle**.

Nota: Falhas repetitivas podem fazer com que o roteador fique alternando entre os estados Idle e Active. Algumas razões para isso podem incluir:

a) As razões descritas no rodapé do slide do estado **Idle**
b) Falha de configuração das sessões.
c) Saturação de Rede.
d) Interface de rede instável.

Estados do protocolo BGP

Quando a sessão se encontra no estado **OpenSent**, obedecerá aos seguintes critérios:

a) Aguardará por uma mensagem **Open** de seu neighbor.
b) Uma vez a mensagem recebida, o roteador irá executar a validação da mensagem **Open**.
c) Se acontecer algum erro, será devido a um dos campos da mensagem **Open** não corresponder entre os neighbors, por exemplo: Versão BGP incompatível, o neighbor espera por um número diferente de AS e etc. O roteador então irá enviar uma mensagem de notificação ao neighbor informando o motivo do erro.
d) Se não houver erros, uma mensagem **Keepalive** será enviada, vários timers serão definidos e o estado será alterado para **OpenConfirm.**

Estados do protocolo BGP

Quando a sessão se encontra no estado **OpenConfirm**, obedecerá aos seguintes critérios:

a) O neighbor está aguardando uma mensagem **Keepalive** de seu parceiro.
b) Se uma mensagem **Keepalive** for recebida e não houver expirado nenhum timer antes da recepção do **Keepalive**, o estado será mudado para **Established**.
c) Caso um dos timers expirem antes da mensagem **Keepalive** ser recebida, ou se ocorrer um erro, o roteador irá alterar o estado para **Idle** novamente.

Estados do protocolo BGP

Quando a sessão se encontra no estado **Established**, a sessão BGP estará concluída e a partir daí obedecerá aos seguintes critérios:

a) Nesse estado, os peers irão trocar mensagens **Update** contendo informações sobre cada rota a ser anunciada para o neighbor (troca de NLRIs).
b) Se houver qualquer erro na mensagem **Update**, então será enviada uma mensagem **Notification** ao peer e a sessão BGP voltará ao estado **Idle.**

Importante: Não confundir o estado **Active** com **Established** pois o estado final que indica que uma sessão está funcional é o estado **Established**. O estado **Active** é justamente o contrário, que indica que uma sessão **não foi estabelecida.**

Atributos BGP

Vamos imaginar um cenário onde recebemos uma determinada rota (vamos pegar como exemplo a rede 8.8.8.0/24 de diversos caminhos (operadoras, IXs ou downstreams). Como o protocolo BGP escolhe qual é o melhor caminho? Quais são os critérios para lhe determinar?

Ao contrário de protocolos IGP **link-state** como o OSPF que levam em consideração coisas como velocidade da interface, o BGP escolhe o melhor caminho através de uma lista de atributos que são validados de cima para baixo numa ordem de desempate.

Atributos BGP

Abaixo estão os atributos BGP e qual a sua prioridade associada:

Weight (1)
Local Preference (2)
Originate (3)
Tamanho do AS_PATH (4)
Código Origin (5)
MED (6)
Preferência de eBGP acima de iBGP (7) *
Caminho mais curto IGP em comparação com next-hop BGP (8)
Caminho mais antigo (9)
Router-ID (10)
Endereço IP do Neighbor (11)

Atributos BGP

Weight

Preferir o caminho com o maior peso (weight). Esse atributo **não é transitivo** e é proprietário Cisco. O valor padrão é 0 para todas as rotas que não são originadas pelo roteador local.

Local Preference

O atributo **local preference** tem relevância local (num mesmo AS) e **é transitivo apenas aos vizinhos iBGP**. Será escolhido o caminho com maior **local-preference.** O valor padrão é 100.

Atributos BGP

Originate

Preferirá o caminho que o roteador local originou, ou seja, que aprendeu por BGP e instalou na tabela. Preferirá esse caminho do que o que roteadores vizinhos tiverem instalado por BGP e redistribuído para o roteador local.

Tamanho do AS_PATH

Preferirá o caminho com o AS_PATH mais curto. Ex: AS_PATH 16735 10429 será preferido em comparação com AS_PATH 10429 262721 262761.

Atributos BGP

Código Origin

Irá preferir o menor código **origin**. Existem três códigos:

a) **IGP**
b) **EGP**
c) **INCOMPLETE**

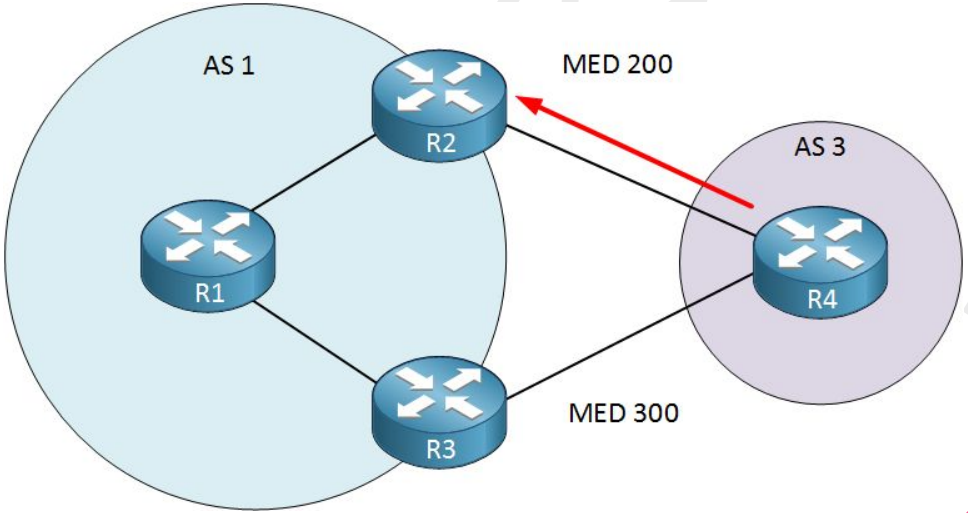
A ordem é **IGP** < **EGP** < **INCOMPLETE**, com efeito, a rota **IGP** será preferida contra uma rota **INCOMPLETE**.

Rotas **INCOMPLETE** são rotas originadas por outros protocolos (como estática ou conectada) e apenas redistribuídas pelo BGP.

Atributos BGP

MED

Será escolhido o caminho com o menor MED. O atributo MED é transitivo entre dois sistemas autônomos porém possui relevância apenas em cenários muito específicos.



Atributos BGP

Preferência de eBGP acima de iBGP (7) *

Preferirá caminho eBGP ao invés de um caminho aprendido via iBGP (nota: apesar de possuírem o mesmo A.D. em juniper, o roteador segue a regra de preferir eBGP acima de iBGP)

Caminho IGP mais curto em detrimento ao next-hop BGP

Preferirá o caminho dentro do mesmo AS com a menor metrica IGP em comparação com ao Next-Hop BGP

Caminho mais antigo

Escolherá o caminho que recebeu primeiro, ou seja, rota mais antiga.

Atributos BGP

Router-ID

Preferirá o caminho (neighbor) cujo roteador tiver o menor Router-ID. O Router-ID por padrão é baseado no IP mais alto ou IP de Loopback, mas pode ser configurado manualmente.

Endereço IP do Neighbor

Preferirá o caminho cujo neighbor tem o IP mais baixo. Se você tem dois roteadores eBGP e dois links entre eles, o **Router-ID** será igual. Nesse caso o **tiebreaker** será o IP de neighbor mais baixo.

Communities BGP

Communities BGP são de longe a extensão mais utilizada do BGPv4 e isto não é à toa, a facilidade que elas agregam faz todo o sentido para que sejam as extensões favoritas.

Communities são como unidades organizacionais **transitivas**, as quais conseguimos adicionar marcações manuais para serem trabalhadas por filtros locais ou remotos (devido à transitividade dessa extensão).

Exemplo: Você pode adicionar uma **community** a todas as rotas recebidas de um **neighbor** (digamos ALGAR) específico e batizá-la de **ALGAR** por exemplo, a partir dali você pode trabalhar em filtros específicos afetando todas as rotas recebidas desse neighbor sem a necessidade de citar rota por rota.

RAZEL

Communities BGP

Diferentemente das **prefix-lists** que precisam de ser preenchidas manualmente, as **communities** podem adicionar ou remover prefixos dinamicamente de acordo com a lógica do filtro que está sendo utilizado.

Os vizinhos BGP podem negociar **communities** entre si que atendam aos mais diversos interesses entre as partes, os exemplos mais clássicos são:

a) Communities de Blackhole
b) Communities para Anuncios Regionais (Anunciar somente internacional, somente nacional, etc).

Communities BGP

Existem dois tipos de **communities**, as **communities padrão** e as **communities estendidas**, essas últimas são as únicas que suportam AS de 32bits porém não estão restritas a isto, possuem outros recursos como por exemplo a marcação **origin** ou **route target**.

O formato das **communities padrão** é: **ASN:IDENTIFICADOR**, ex: **16735:666**

O formato das **communities estendidas** é: **ORIGIN/TARGET:ASN:IDENTIFICADOR**, ex: **target:16765:666**.

Quando for usado um AS de 32bits é necessário usar uma flag chamada Larger **(L)** no AS citado na community, ex: **target:262761L:666**

Configurações de sessões eBGP e iBGP

EBGP (v4 sem multihop)

Antes de se configurar uma sessão eBGP, deve-se ter à disposição os seguintes dados:

a) **IP diretamente conectado (ou multihop, habilitando a flag)**
b) **Neighbor**
c) **AS remoto (peer as)**
d) **Se irá ou não utilizar chave de segurança**

Configurações de sessões eBGP e iBGP

Finalmente iremos realizar a configuração da sessão (toda a configuração fica no nível de hierarquia *"protocols bgp"* e para cada AS remoto será necessária a criação de um *"grupo BGP"* distinto. Ex:

[edit protocols bgp group IPV4-ALGAR]
set **type external** # Seleciona o **eBGP**
set **family inet unicast** # Informa que estará utilizando ipv4 e falará unicast apenas
set **neighbor** <X.Y.W.Z> # Informa qual é o IP do neighbor BGP
set **peer-as** <XXXXX> # Informa o AS com o qual se deseja o peering
set **import** <import policy> # Seleciona o filtro de import
set **export** <export policy> # Seleciona o filtro de export
set **local-address** <X.Y.W.Z> # Define o endereço local
set **local-as** <X.Y.W.Z> # Define o AS local
set **authentication-key** <chave md5 # Define a chave md5

Configurações de sessões eBGP e iBGP

Apesar do exemplo anterior possuir diversos parâmetros de sessão, apenas os itens em vermelho são mandatórios, todos os outros são opcionais.

Em versões mais recentes do JunOS, assim como nos roteadores Cisco, há relatos de que não há necessidade de especificar tipo de BGP (i ou e) pois o próprio sistema entenderá que ASNs iguais serão sempre iBGP e todo o resto, eBGP.

A família de protocolo determina qual “sabor” de NLRI será trocado e apesar do roteador tentar identificar isso, não é infalível pois há meios de se fechar uma sessão BGP com um neighbor IPv4 e por ela receber rotas IPv6 (e esse é só um exemplo de problemática possível).

Configurações de sessões eBGP e iBGP

eBGP (v6 sem multihop)

Para IPv6 a única mudança que iremos ter de padrão é a troca de família para inet6.

[edit protocols bgp group IPV6-ALGAR]

set **type external** # Seleciona o **eBGP**

set **family inet6 unicast** # Informa que estará utilizando ipv4 e falará unicast apenas

set **neighbor** <X:X:X:X:X:X:X:X> # Informa qual é o IP do neighbor BGP

set **peer-as** <X:X:X:X:X:X:X:X> # Informa o AS com o qual se deseja o peering

set **import** <import policy> # Seleciona o filtro de import

set **export** <export policy> # Seleciona o filtro de export

set **local-address** <X:X:X:X:X:X:X:X> # Define o endereço local

set **local-as** <X:X:X:X:X:X:X:X> # Define o AS local

set **authentication-key** <chave md5 # Define a chave md5

Configurações de sessões eBGP e iBGP

eBGP Multihop

Para multihop, acrescentaremos a linha **multihop ttl**.

[edit protocols bgp group IPV4-ALGAR]

set **type external** # Seleciona o **eBGP**

set **family inet unicast** # Informa que estará utilizando ipv4 e falará unicast apenas

set **neighbor** <X.Y.W.Z> # Informa qual é o IP do neighbor BGP

set **peer-as** <XXXXX> # Informa o AS com o qual se deseja o peering

set **import** <import policy> # Seleciona o filtro de import

set **export** <export policy> # Seleciona o filtro de export

set **local-address** <X.Y.W.Z> # Define o endereço local

set **local-as** <X.Y.W.Z> # Define o AS local

set **authentication-key** <chave md5 # Define a chave md5

set **multihop ttl 255**

Configurações de sessões eBGP e iBGP

iBGP

Antes de se configurar uma sessão iBGP, deve-se ter à disposição os seguintes dados:

a) **Neighbor**
b) **Se irá ou não utilizar chave de segurança**

OBS: O JunOS cria adjacências IGP sob todas as subredes quando o IGP está configurado em uma interface lógica; esse comportamento é importante de se notar porque outros vendors formam adjacência somente através do enddereço primário de uma interface.

Configurações de sessões eBGP e iBGP

iBGP Da mesma forma, para um roteador dentro do mesmo AS, pode-se utilizar o exemplo abaixo:

[edit protocols bgp group IPV4-ALGAR]
set **type internal** # Seleciona o **iBGP**
set **family inet unicast** # Informa que estará utilizando ipv4 e falará unicast apenas
set **neighbor** <X.Y.W.Z> # Informa qual é o IP do neighbor BGP
~~set **peer-as** <XXXXX> # Informa o AS com o qual se deseja o peering~~
set **import** <import policy> # Seleciona o filtro de import
set **export** <export policy> # Seleciona o filtro de export
set **local-address** <X.Y.W.Z> # Define o endereço local
~~set **local-as** <X.Y.W.Z> # Define o AS local~~
set **authentication-key** <chave md5 # Define a chave md5

RAZEL

Configurações de sessões eBGP e iBGP

Repare que em iBGP, ao contrário do eBGP, não é mais necessário especificar **peer-as** e **local-as**, pois fica evidente a utilização do mesmo AS em ambos locais.

Todavia para não ser obrigado a informar o **local-as** na sessão, deve-se primeiro especificar o AS global da caixa com o comando **set routing-options autonomous-system XXXXX.**

Roteadores Juniper, ao contrário dos cisco IOS, podem executar múltiplos AS na mesma “caixa”, entretanto, o AS que definimos em routing-options será usado uma vez que não haja **local-as** especificado na sessão BGP que está a ser configurada.

Filtros BGP

Há diferenças entre a aplicação de peering entre parceiros que estarão IMPORTANDO nossas rotas e EXPORTANDO para o mundo, que nos fornecem trânsito, tal como Algar, etc. e as que estão EXPORTANDO as rotas delas e IMPORTANDO nossa tabela de rotas.

Filtros BGP

2. Configurar Peering BGP Full Route com parceiro Juniper (Clientes de Trânsito/Sistemas Autônomos)

Quando fornecemos trânsito a outro parceiro devemos ter em mente a imagem do tipo de circuito que será entregue e passar ao cliente as informações necessárias para o peering, entretanto, os filtros serão diferentes.

Devemos fornecer ao cliente:

ASN, Neighbor IP, IP WAN, Vlan (opcional)

Para isto, devemos primeiro configurar o nosso lado da sessão, como o exemplo a seguir:

Filtros BGP

A configuração de Filtros BGP é FUNDAMENTAL para elaboração de um peering confiável, uma vez que, se forem exportados prefixos em demasiado, isto poderia causar loop em um peering redundante tal como exige o PTT METRO.
Nos filtros, devemos aceitar, rejeitar ou descartar o filtro criado com os comandos accept, reject e discard, em termos separados.
Ex: term 1 especifica o tipo de tráfego a ser filtrado term 2 (do mesmo filtro) aceita, rejeita ou descarta o tráfego descrito no term 1 daquele mesmo filtro.

Filtros BGP

Filtros de IMPORT:

Devemos filtrar no IMPORT somente as rotas que desejamos RECEBER e no EXPORT as rotas que desejamos anunciar ao parceiro.

Podemos utilizar como base o exemplo abaixo:

set policy-options policy-statement NOME-DA-POLICY-DE-IMPORT term NUMERO-DO-TERMO from protocol NOME-DO-PROTOCOLO (aqui filtramos para importar somente as rotas do protocolo que desejamos:

Filtros BGP

Exemplo:

Caso apenas seja interessante importarmos do parceiro rotas anunciadas via protocolo BGP
(que é o caso das operadoras) e rejeitar o restante podemos utilizar:

```
set policy-options policy-statement BGP-IMPORT-VIVO-V4 term 1 from protocol bgp
set policy-options policy-statement BGP-IMPORT-VIVO-V4 term 1 then local-preference 500
set policy-options policy-statement BGP-IMPORT-VIVO-V4 term 1 then accept
set policy-options policy-statement BGP-IMPORT-VIVO-V4 term 2 then reject
```

Este é um filtro bastante utilizado para peering com operadoras

Filtros BGP

Local Preference

É possível especificar qual é o peso daquele peering com a rede num todo, exemplo, rede com maior capacidade (Link) ter preferência acima de outra rede que possui link menor.

Na local-preference pode ser especificado qualquer valor entre 0 e 1000 e pode ajudar a fazer uma “espécie” de Load-Balancing, dando preferência a utilizar o circuito com maior link.

Nas boas práticas, devemos sempre especificar o local-preference ao efetuar peering com operadoras.

Filtros BGP

Filtros de Export:

A configuração de filtros de EXPORT é necessária para efetuarmos os anúncios corretamente, evitando assim o anúncio de LIXO para o mundo, o que pode causar sérios problemas.

Novamente é possível filtrar por tipo de protocolo, por prefixos, etc.

Podemos tomar como base os exemplos abaixo:

set policy-options policy-statement NOME-DO-FILTRO-DE-EXPORTAÇÃO term NUMERO-DO-TERMO from protocol NOME-DO-PROTOCOLO

set policy-options policy-statement NOME-DO-FILTRO-DE-EXPORTAÇÃO term NUMERO-DO-TERMO from route-filter IPV4/IPV6/NOTAÇÃO CIDR exact/longer/orlonger/prefix-length-range/through/upto

Filtros BGP

No caso de filtragem por protocolo podemos tomar o exemplo abaixo (não recomendado):
Caso seja de nosso interesse anunciar apenas os prefixos que são recebidos por BGP (seja iBGP ou eBGP) de parceiros e/ou nós de rede, podemos filtrá-lo por protocolo, entretanto, se há outro parceiro que esteja anunciando BGP Full Route além deste onde estão sendo feitos os anúncios, isto iria tentar anunciar para o “mundo” todas as rotas recebidas do “mundo” causando um loop muito prejudicial. Este tipo de exemplo é utilizado para peering com uma única operadora, pois a opção advertise-peer-as que por padrão no Juniper vem desativada previne que seja anunciado ao mesmo parceiro as rotas que ele mesmo envia.

NUNCA deve se habilitar a opção advertise-peer-as numa situação como esta.

Diferenças entre Juniper e outros vendors BGP

Alguns vendors tem uma distância administrativa *(AD) (preference)* de **20** para EBGP e **200** para IBGP. A plataforma Juniper usa o mesmo valor **170** tanto para EBGP quanto IBGP. Entretanto não há impacto operacional pois o JunOS sempre prefere rotas EBGP acima das IBGP.

Outra diferença é na diferença da AD do IGP comparado a distancia do BGP. Por exemplo, alguns vendors designam uma distancia de 110 para as rotas OSPF. Esse número é maior que a distância EBGP de 20 e resulta na preferência da rota EBGP em vez da rota OSPF.

A correção desse comportamento deve ser realizada utilizando o recurso **preference** explicitamente (setando o AD manualmente)
na sua configuração BGP.

Dúvidas?

www.razel.com.br
(17) 99629-7017
caio@razel.co