CURSO IPv6 BÁSICO
ROTEAMENTO

Rodrigo Regis dos Santos
Antônio M. Moreiras
Eduardo Ascenço Reis
Ailton Soares da Rocha

Núcleo de Informação e Coordenação do ponto BR

São Paulo
2011

Núcleo de Informação e Coordenação do Ponto BR

Diretor Presidente
Demi Getschko

Diretor Administrativo
Ricardo Narchi

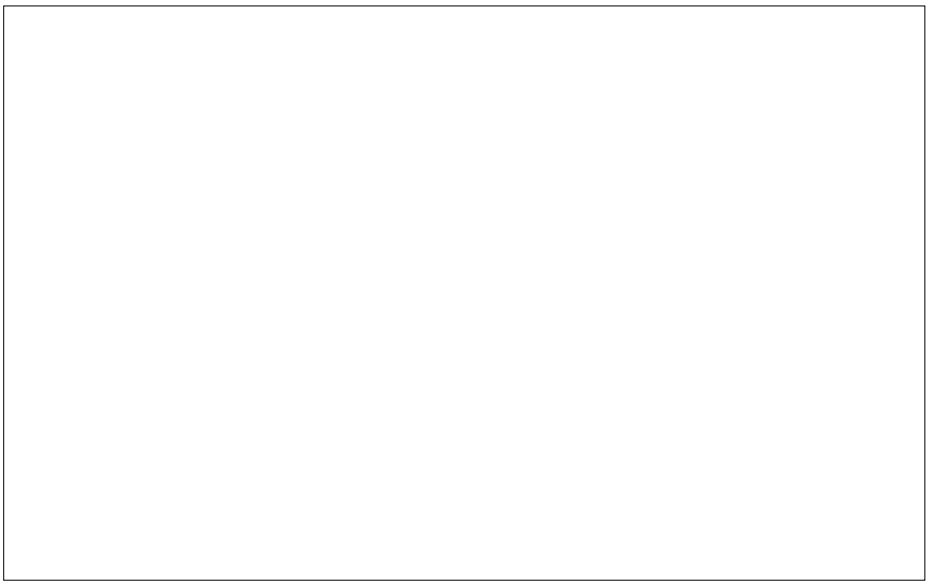
Diretor de Serviços
Frederico Neves

Diretor de Projetos Especiais e de Desenvolvimento
Milton Kaoru Kashowakura

Centro de Estudos e Pesquisas em Tecnologia de Redes e Operações – CEPTRO.br
Antônio Marcos Moreiras

Coordenação Executiva e Editorial: Antônio Marcos Moreiras

Autores / Design / Diagramação
Rodrigo Regis dos Santos
Antônio Marcos Moreiras
Eduardo Ascenço Reis
Ailton Soares da Rocha



Sobre o Projeto IPv6.br

O IPv6 é a nova geração do Protocolo Internet.

Ele já vem sendo utilizado há algum tempo. Mas agora sua implantação deve ser acelerada. Ela é imprescindível para a continuidade do crescimento e da evolução da Internet!

O objetivo do projeto IPv6.br do NIC.br é estimular a utilização do novo protocolo na Internet e nas redes brasileiras. Para saber mais acesse o sítio Internet **www.ipv6.br** ou entre em contato pelo e-mail **ipv6@nic.br**.

O **CEPTRO.br**- Centro de Estudos e Pesquisas em Tecnologia de Redes e Operações do NIC.br – é responsável por projetos que visam melhorar a qualidade da Internet no Brasil e disseminar seu uso, com especial atenção para seus aspectos técnicos e de infraestrutura. Mais informações podem ser obtidas no sítio Internet **www.ceptro.br**.

Sobre os autores

Rodrigo Regis dos Santos é Bacharel em Ciência da Computação pela Universidade Presbiteriana Mackenzie e atualmente está especializando-se em Gestão e Infraestrutura de Telecomunicações pela mesma Universidade. Especialista em IPv6, trabalha no NIC.br atuando como um dos responsáveis pelo projeto IPv6.br, que tem por objetivo incentivar o uso do protocolo no país.

Antonio M. Moreiras é engenheiro eletricista e mestre em engenharia pela POLI/USP, MBA pela UFRJ, além de ter estudado Governança da Internet na Diplo Foundation e na SSIG 2010. Trabalha atualmente no NIC.br, onde está envolvido em projetos para o desenvolvimento da Internet no Brasil, como a disseminação do IPv6, a disponibilização da Hora Legal Brasileira via NTP, estudos da Web, ENUM, entre outros.

Eduardo Ascenço Reis é especialista em redes IP, sistemas Unix e serviços Internet. Como formação, possui Curso de Especialização em Redes de Computadores pelo LARC/USP e Bacharelado em Ciências Biológicas pela USP. Sua atuação profissional em TI ocorre desde 1995, com experiências nas empresas: Universidade de São Paulo (USP), Ericsson Brazil, comDominio (IDC - AS16397), CTBC Multimídia (NSP, ISP - AS27664). Atualmente atua como supervisor de projetos no Centro de Estudos e Pesquisas em Tecnologia de Redes e Operações (CEPTRO.br) do Núcleo de Informação e Coordenação do Ponto BR (NIC.br) e é um dos responsáveis pelo PTTmetro (PTT.br).

Ailton Soares da Rocha é Analista de Projetos no NIC.br, onde está envolvido com pesquisas e projetos relacionados à infraestrutura da Internet no país, com importante atuação nos projetos IPv6.br e NTP.br. Engenheiro eletricista e de telecomunicações graduado pelo INATEL, atuou por mais de 10 anos na coordenação da área de redes e Internet da instituição.

IPv6.br

IPv6.br

Sobre a licença

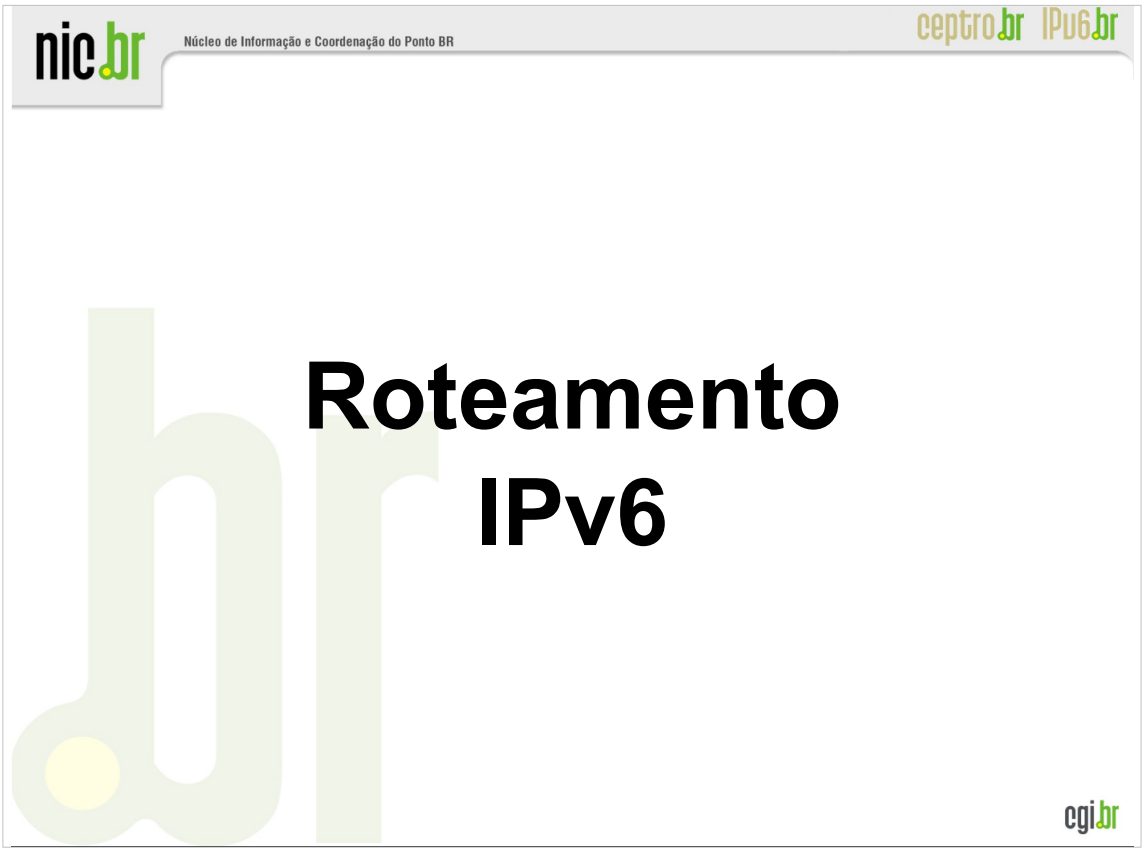
nic.br Núcleo de Informação e Coordenação do Ponto BR

ceptro.br IPv6.br

IPv6.br

A Nova Geração do Protocolo Internet

cgi.br

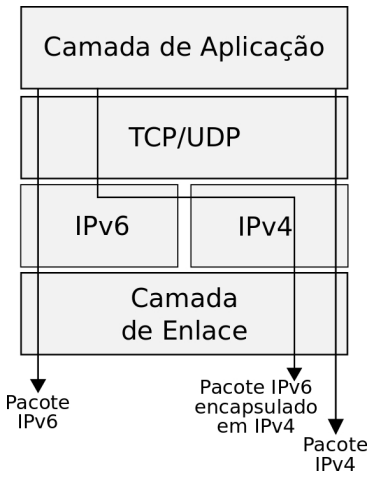


Neste módulo apresentaremos algumas características básicas sobre o funcionamento dos mecanismos de roteamento, tanto interno (IGP) quanto externo (EGP), sempre destacando as principais mudanças em relação ao IPv6. Serão abordados os protocolos de roteamento RIP, OSPF, IS-IS e BGP.

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

Considerações Importantes

- IPv4 e IPv6 → Camada de Rede
- Duas redes distintas
  - Planejamento
  - Suporte
  - *Troubleshooting*
  - Arquitetura dos equipamentos
  - ...

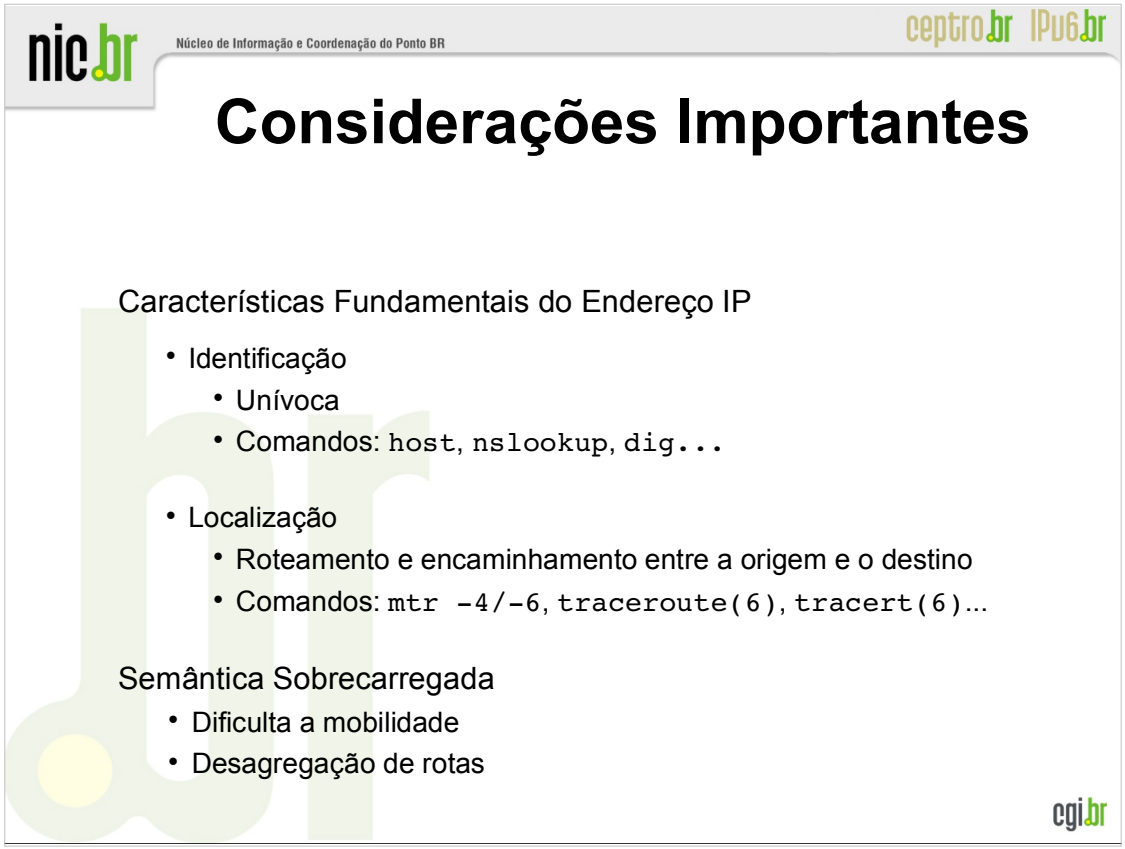


cgi.br

O IPv4 e o IPv6 são protocolos da Camada de Rede, de modo que esta é a única camada diretamente afetada com a implantação do IPv6, sem a necessidade de alterações no funcionamento das demais.

Porém, é preciso compreender que são duas Camadas de Rede distintas e independentes. Isto implica em algumas considerações importantes:

- Como atuar no planejamento e estruturação das redes:
  - Migrar toda a estrutura para Pilha Dupla; migrar apenas áreas críticas; manter duas estruturas distintas, uma IPv4 e outra IPv6; etc.
  - Em redes com Pilha Dupla algumas configurações devem ser duplicadas como DNS, *firewall* e protocolos de roteamento.
- No suporte e resolução de problemas será preciso detectar se há falhas na conexão da rede IPv4 ou da rede IPv6;
- Novos equipamentos e aplicações precisam ter suporte às funcionalidades dos dois protocolos.

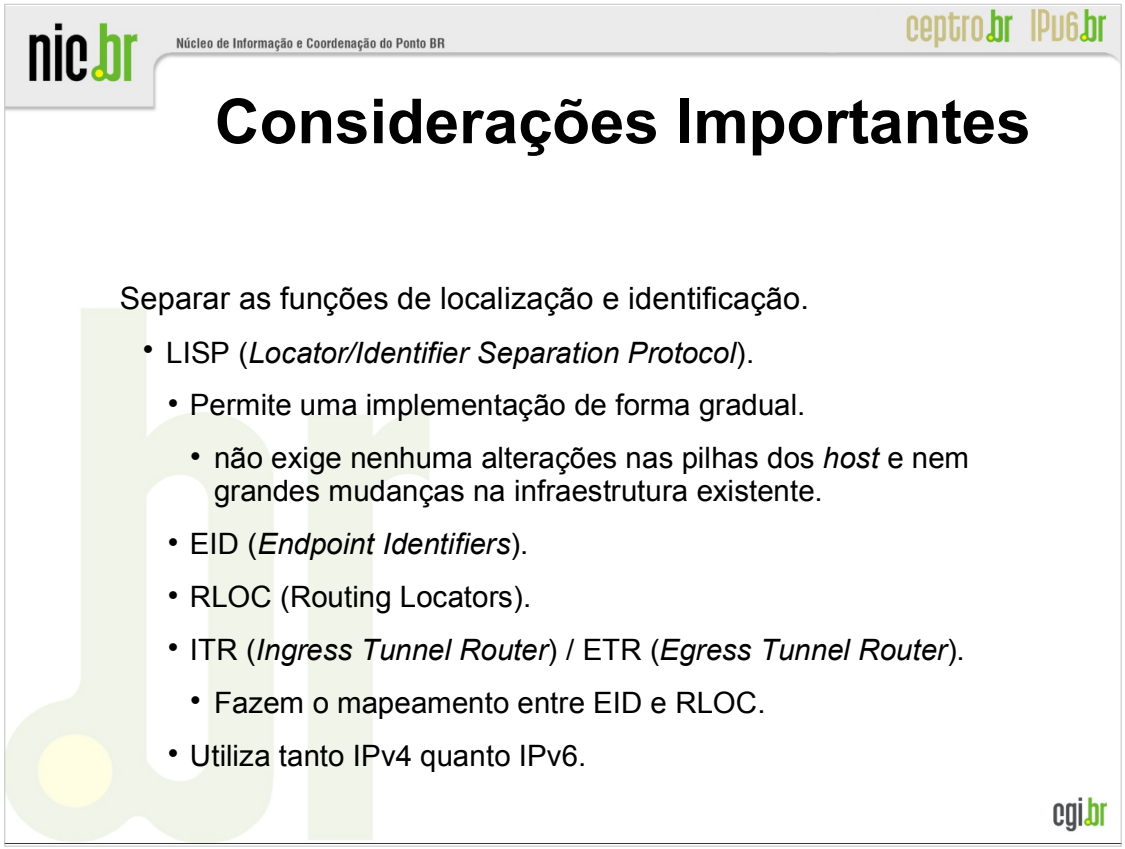


A Camada de Rede está associada principalmente a duas características:

- **Identificação** – deve garantir que cada dispositivo da rede seja identificado de forma unívoca, sem chance de erro. Isto é, o endereço IP deve ser único no mundo. Utilizando o comando host, em plataformas UNIX, ou nslookup, em plataformas Windows, pode-se ver a identificação de um serviço, por exemplo. Em redes com Pilha Dupla, um nó será identificado pelos dois endereços.

- **Localização** - indica como chegar ao destino, tomando as decisões de encaminhamento dos pacotes baseando-se no endereçamento, ocorrendo da mesma forma tanto em IPv4 quanto em IPv6. Podemos verificar esta funcionalidade utilizando comandos como *mtr -4* e *-6*, ou *traceroute (traceroute6)*, ou *tracert (tracert6)*. Estes comandos mostram a identificação e a localização de um nó.

A união dessas duas características na Camada de Rede torna a semântica do endereço IP sobrecarregada. Isto implica em questões como a desagregação de rotas, agravando o problema do crescimento da tabela de roteamento global. Uma forma de impedir isso é separar as funções de localização e identificação.



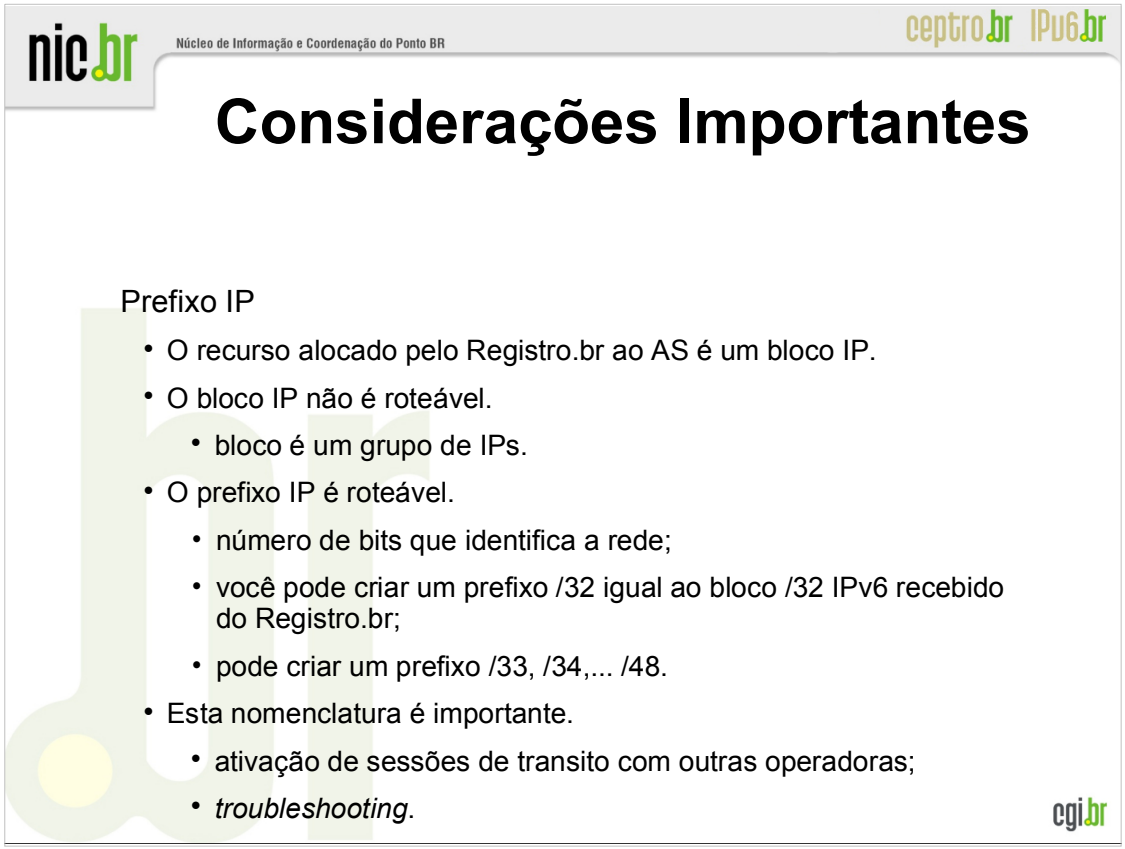
Existe um grupo de trabalho no IETF que discute uma forma de separar essas duas funções (identificação e localização). O LISP (*Locator/Identifier Separation Protocol*) é um protocolo simples que busca separar os endereços IP em *Endpoint Identifiers* (EIDs) e *Routing Locators* (RLOCs). Ele não exige nenhuma alteração nas pilhas dos *host* e nem grandes mudanças na infraestrutura existente, podendo ser implementado em um número relativamente pequeno de roteadores.

Seus principais elementos são:

- *Endpoint ID* (EID): um identificador de 32 bits (para IPv4) ou 128 bits (para IPv6) usado nos campos de endereço de origem e destino do primeiro cabeçalho LISP (mais interno) de um pacote. O *host* obtém um EID de destino da mesma forma que obtém um endereço de destino hoje, por exemplo através de uma pesquisa de DNS. O EID de origem também é obtido através dos mecanismos já existentes, usados para definir o endereço local de um *host;*
- *Routing Locator* (RLOC): endereço IPv4 ou IPv6 de um ETR (*Egress Tunnel Router*). RLOCs são numerados a partir de um bloco topologicamente agregado, e são atribuídos a uma rede em cada ponto em que haja conexão com a Internet global;
- *Ingress Tunnel Router* (ITR): roteador de entrada do túnel que recebe um pacote IP (mais precisamente, um pacote IP que não contêm um cabeçalho LISP), trata o endereço de destino desse pacote como um EID e executa um mapeamento entre o EID e o RLOC. O ITR, em seguida, anexa um cabeçalho "IP externo" contendo um de seus RLOCs globalmente roteáveis, no campo de endereço de origem, e um RLOC, resultado do mapeamento, no campo de endereço de destino;
- *Egress Tunnel Router* (ETR): roteador de saída do túnel que recebe um pacote IP onde o endereço de destino do cabeçalho "IP externo" é um de seus RLOCs. O roteador retira o cabeçalho externo e encaminha o pacote com base no próximo cabeçalho IP encontrado.

Mais informações

- *Locator/ID Separation Protocol (LISP)* - http://www.ietf.org/id/draft-ietf-lisp-06.txt
- *LISP Networking: Topology, Tools, and Documents* - http://www.lisp4.net (apenas conexões IPv4)
- *LISP Networking: Topology, Tools, and Documents* - http://www.lisp6.net (apenas conexões IPv6)

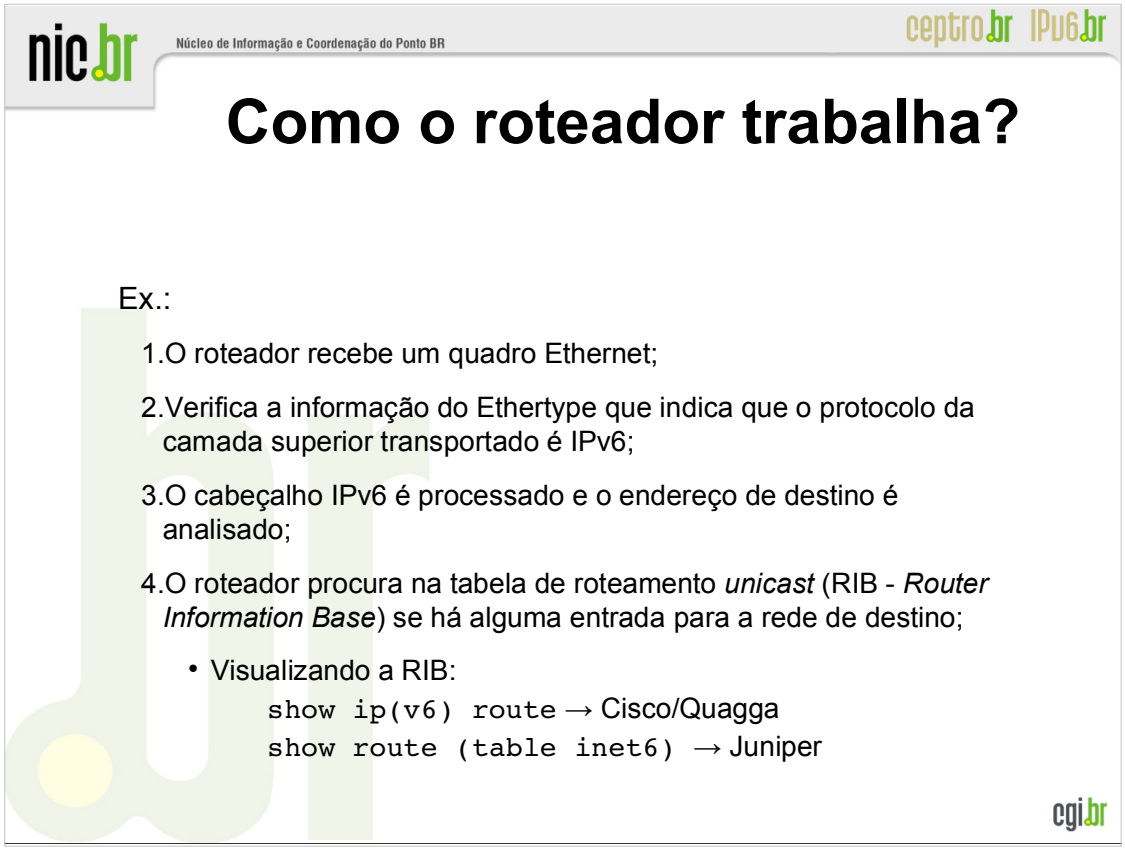


Um definição importante é a de prefixo IP.

O recurso alocado pelo Registro.br aos ASs é um bloco IP, que representa um grupo de endereços IP. O bloco não é um elemento roteável, o que é roteável é o prefixo. O que é possível, por exemplo, é criar um prefixo IPv6 /32 igual ao bloco /32 recebido do Registro.br e anunciar esse prefixo na tabela de rotas. Porém também é possível criar prefixos /33, /34, /48 etc a partir do bloco recebido.

O prefixo representa o número de bits de um endereço que identifica a rede.

Apesar de ser apenas mais uma nomenclatura, essa definição é importante na hora de enviar informações para ativar sessões de trânsito com outras operadoras e na detecção de problemas de conectividade.



Também é importante compreender o funcionamento básico de um roteador, de que forma ele processa os pacotes recebidos e efetua as decisões de encaminhamento. Analise o seguinte exemplo:

- O roteador recebe um quadro Ethernet através de sua interface de rede;
- Verifica a informação do Ethertype, que indica que o protocolo da camada superior transportado é IPv6;
- O cabeçalho IPv6 é processado e o endereço de destino é analisado;
- O roteador procura na tabela de roteamento *unicast* (RIB - *Router Information Base*) se há alguma entrada para a rede de destino;
-

Visualizando a RIB IPv6:

Cisco/Quagga → `show ipv6 route`
Juniper → `show route table inet6`

Visualizando a RIB IPv4:

Cisco/Quagga → `show ip route`
Juniper → `show route`

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

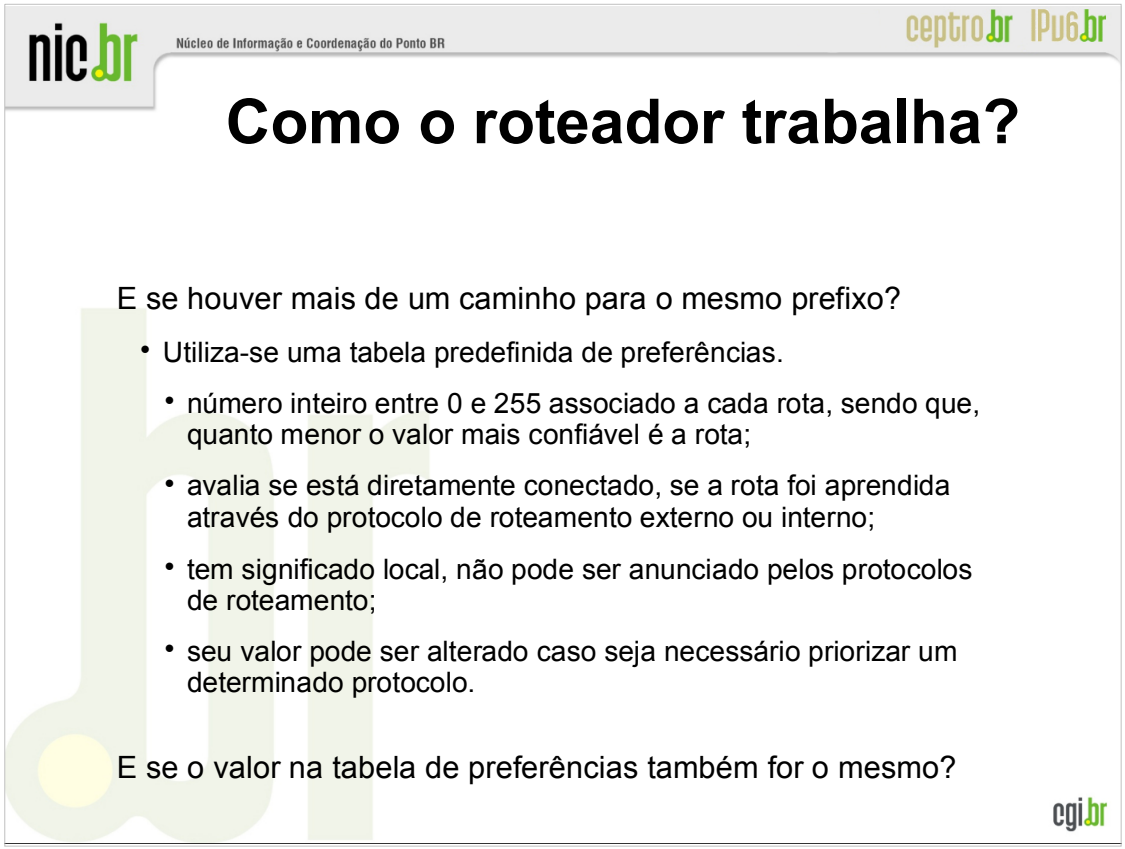
Como o roteador trabalha?

5. *Longest Match* - procura a entrada mais específica. Ex.:
- O IP de destino é 2001:0DB8:0010:0010::0010
- O roteador possui as seguintes informações em sua tabela de rotas:
  - 2001:DB8::/32 via interface A
  - 2001:DB8::/40 via interface B
  - 2001:DB8:10::/48 via interface C
- Os três prefixos englobam o endereço de destino, porém o roteador sempre irá preferir o mais específico, neste caso, o /48;
- Qual é a entrada mais específica IPv4 e IPv6?

6. Uma vez identificado o prefixo mais específico, o roteador decrementa o *Hop-Limit*, monta o quadro Ethernet de acordo a interface, e envia o pacote.

cgi.br

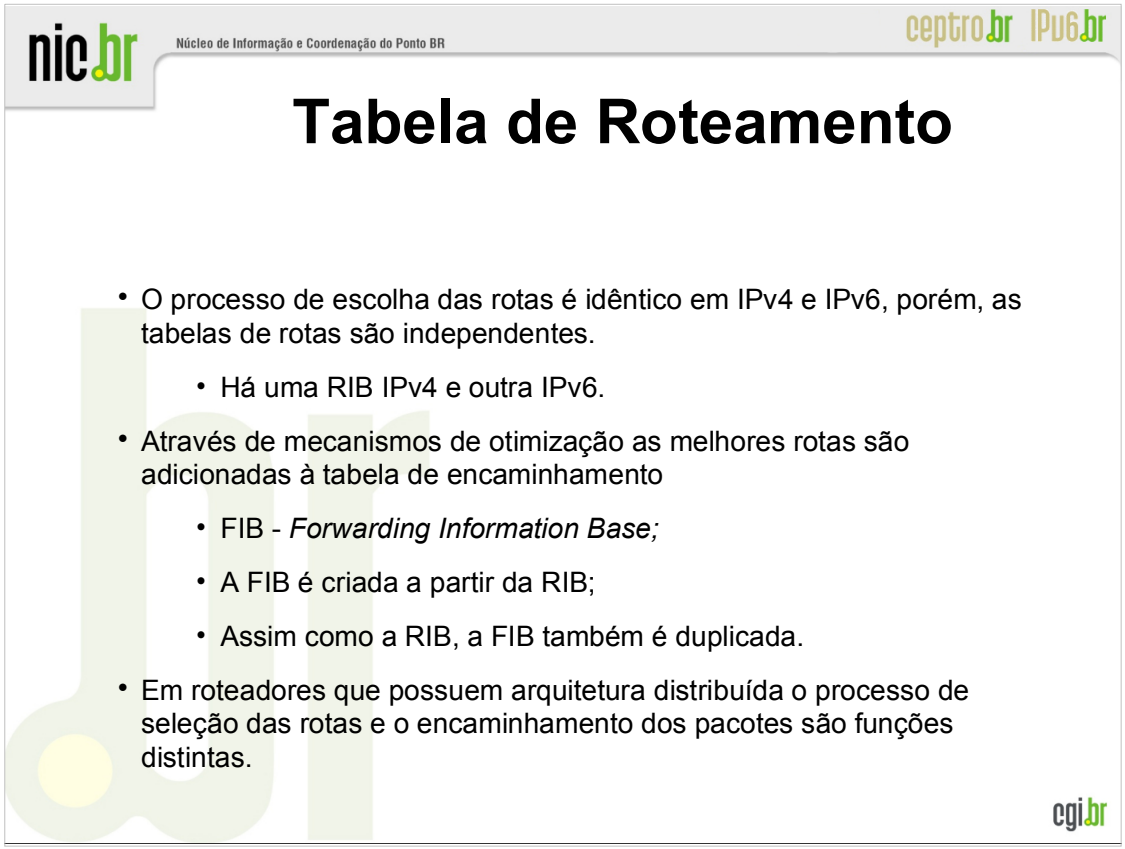
- *Longest Match* - procura a entrada mais específica. Ex.:
  - O IP de destino é 2001:0DB8:0010:0010::0010
  - O roteador possui as seguintes informações em sua tabela de rotas:
  - 2001:DB8::/32 via interface A
  - 2001:DB8::/40 via interface B
  - 2001:DB8:10::/48 via interface C
- Os três prefixos englobam o endereço de destino, porém o roteador sempre irá preferir o mais específico, neste caso, o /48;
- Uma vez identificado o prefixo mais específico, o roteador decrementa o *Hop-Limit*, monta o quadro Ethernet de acordo a interface, e envia o pacote.



Se o roteador localizar mais de uma caminho para o mesmo destino e com o mesmo valor de *longest match*, ele utilizará uma tabela predefinida de preferências (conceito de *Distância Administrativa* da Cisco).

Os valores desta tabela são números inteiros entre 0 e 255 associados a cada rota, sendo que quanto menor o valor mais confiável é a rota. Os valores são atribuídos avaliando se a rota está diretamente conectada, se foi aprendida através do protocolo de roteamento externo ou interno, etc. Estes valores têm significado apenas local, não podendo ser anunciados pelos protocolos de roteamento, e caso seja necessário, podem ser alterados para priorizar um determinado protocolo.

Caso também seja encontrado um mesmo valor na tabela de preferências, há equipamentos e implementações que, por padrão, realizam balanceamento de tráfego.



O processo de escolha das rotas é idêntico em IPv4 e IPv6, porém, as tabelas de rotas são independentes. Por exemplo: há uma RIB IPv4 e outra IPv6.

Para otimizar o envio dos pacotes, existem mecanismos que adicionam apenas as melhores rotas a uma outra tabela, a tabela de encaminhamento (FIB - *Forwarding Information Base*). Um exemplo deste mecanismo é o CEF (*Cisco Express Forwarding*) da Cisco.

A FIB é criada a partir da RIB, e assim como a RIB, ela também é duplicada se a rede estiver configurada com Pilha Dupla. Com isso, há mais informações para serem armazenadas e processadas.

Em roteadores que possuem arquitetura distribuída, o processo de seleção das rotas e o encaminhamento dos pacotes são funções distintas.

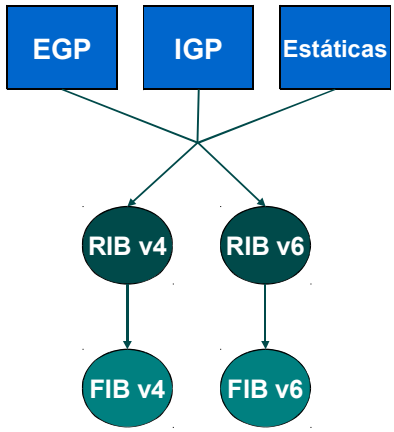
Ex.:

- Roteadores 7600 da Cisco, a RIB reside no módulo central de roteamento e a FIB nas placas das interfaces.
- Roteadores Juniper da série M, a *Router Enginie* é a responsável pela RIB, e a FIB também reside nas placas das interfaces (*Packet Forwarding Engine* - PFE).

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

Tabela de Roteamento

- São as informações recebidas pelos protocolos de roteamento que "alimentam" a RIB que por sua vez "alimenta" a FIB.
- Os Protocolos de Roteamento se dividem em dois grupos:
  - **Interno (IGP)** - protocolos que distribuem as informações dos roteadores dentro de Sistemas Autônomos. Ex.: OSPF; IS-IS; RIP.
  - **Externo (EGP)** - protocolos que distribuem as informações entre Sistemas Autônomos. Ex.: BGP-4.



cgi.br

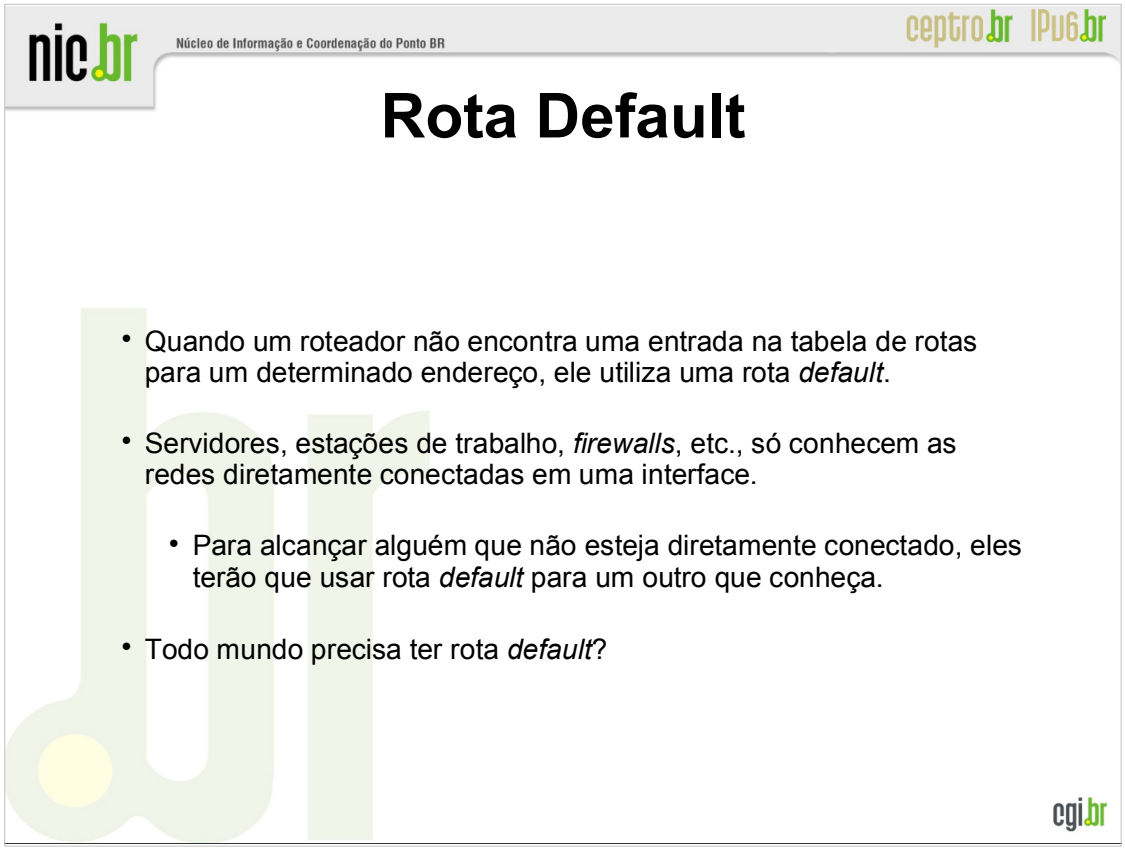
É o mecanismo de roteamento que possibilita o encaminhamento de pacotes de dados entre quaisquer dois dispositivos conectados à Internet.

Para atualizar as informações utilizadas pelos roteadores para encontrar o melhor caminho disponível no encaminhamento dos pacotes até o seu destino, utilizam-se os protocolos de roteamento. São as informações recebidas pelos protocolos de roteamento que "alimentam" a RIB, que por sua vez "alimenta" a FIB.

Estes protocolos se dividem em dois grupos:

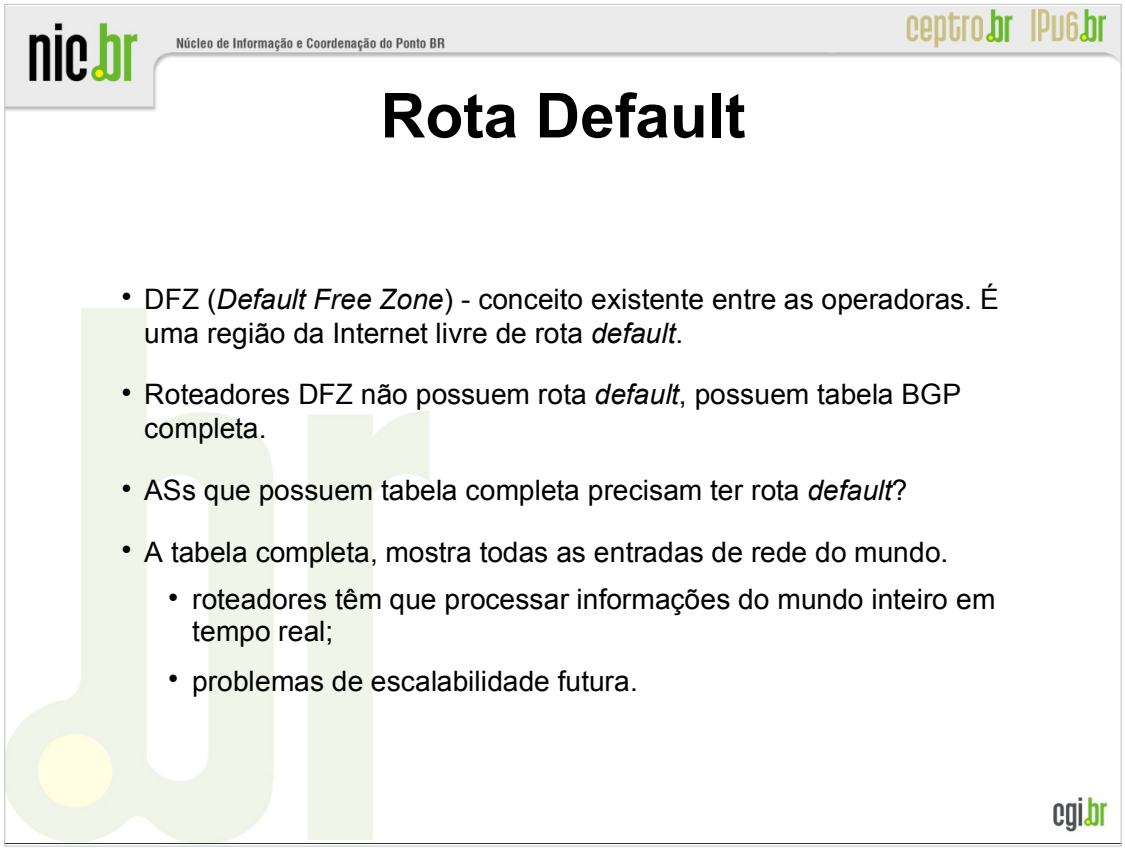
- Interno (IGP) - protocolos que distribuem as informações dos roteadores dentro de Sistemas Autônomos. Como exemplo desses protocolos podemos citar: OSPF; IS-IS; e RIP.

- Externo (EGP) - protocolos que distribuem as informações entre Sistemas Autônomos. Como exemplo podemos citar o BGP-4.



Caso o roteador receba um pacote cujo o endereço de destino não esteja explicitamente listado na tabela de rotas, ele utilizará sua rota *default.*

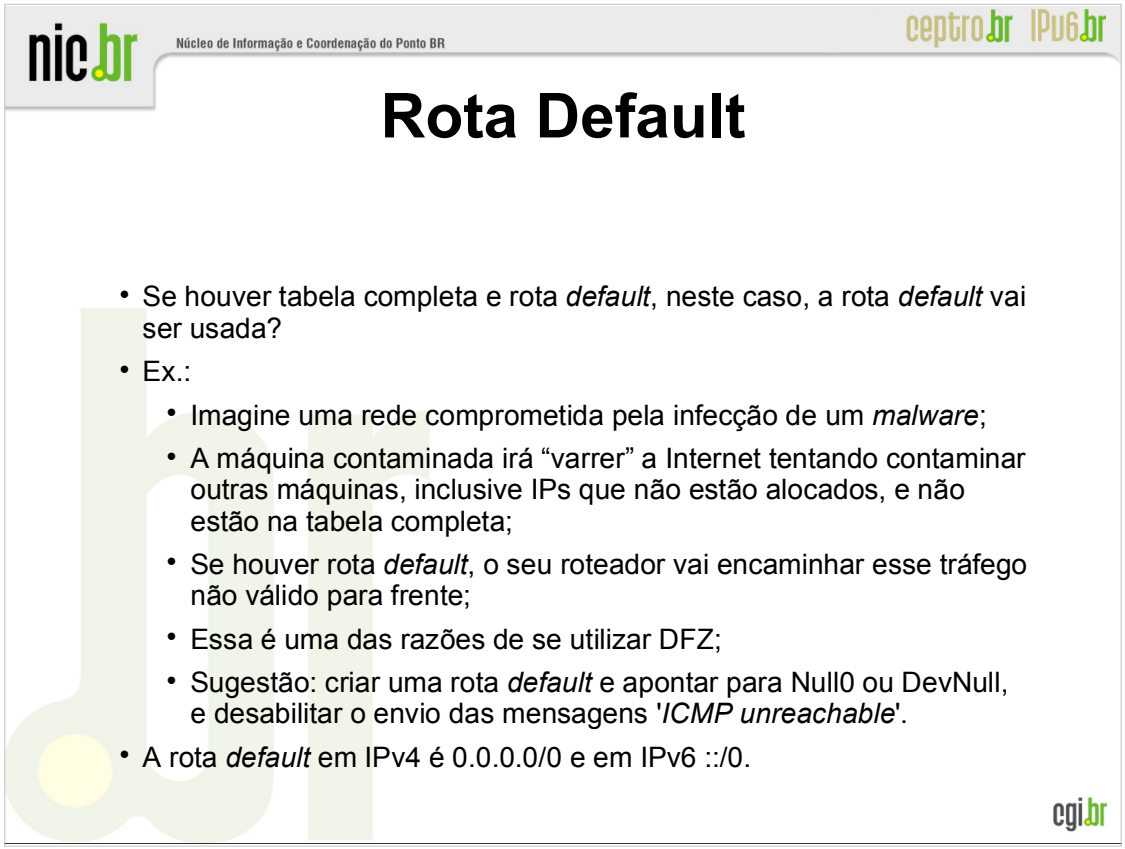
*S*ervidores e estações de trabalho, naturalmente precisam de uma rota *default*. Eles não são equipamentos de rede, eles só conhecem as redes diretamente conectadas em suas interfaces. Se eles quiserem alcançar alguém que não esteja diretamente conectado, eles terão que usar rota *default* para um outro equipamento que conheça.



Existe um conceito entre as operadoras que delimita uma região da Internet livre de rota *default, a* DFZ (*Default Free Zone*).

Um AS que possua tabela completa não precisa ter rota *default*, pois a tabela completa, mostra todas as entradas de rede do mundo.

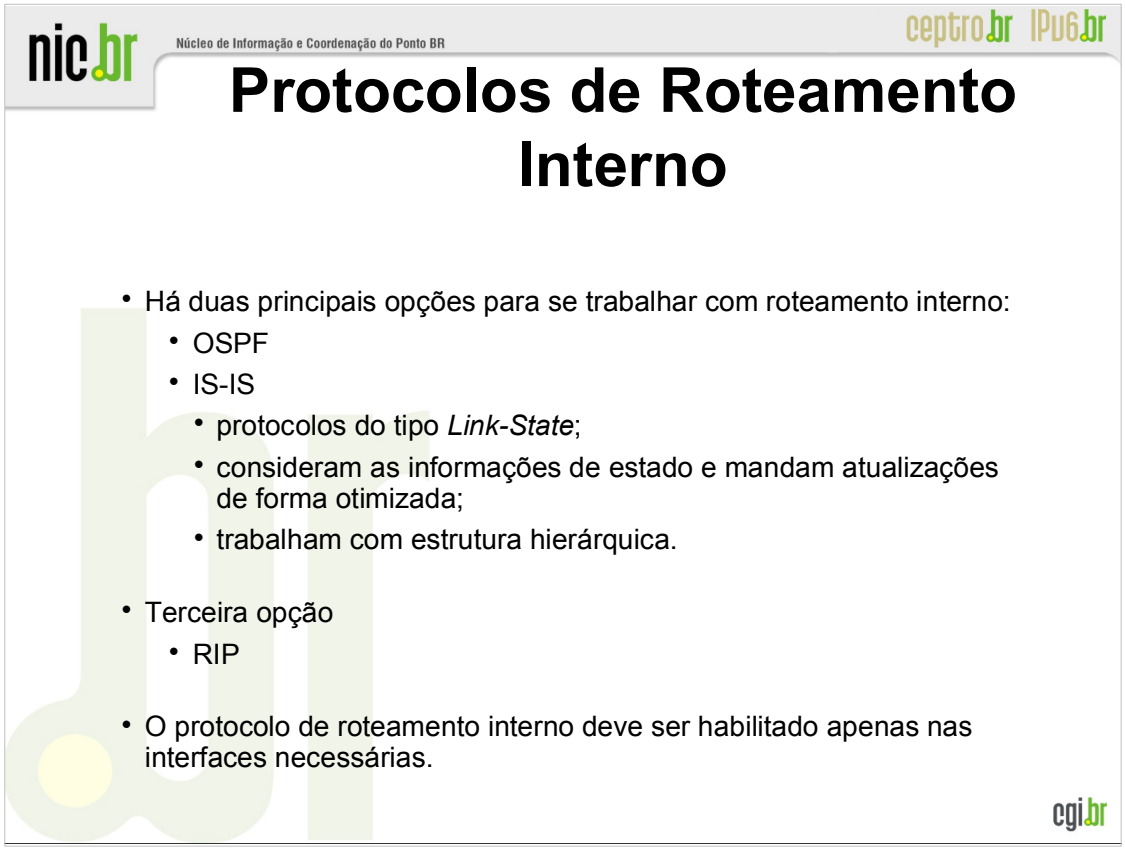
Esse modelo é bom e funcional, porém, isso pode acarretar alguns problemas. Os roteadores têm que processar informações do mundo inteiro em tempo real; e há também problemas de escalabilidade futura.



A utilização de rota *default* por roteadores que possuam tabela completa pode ocasionar alguns problemas.

Imagine a seguinte situação como exemplo: uma rede foi comprometida pela infecção de um *malware*. A máquina contaminada irá "varrer" a Internet tentando contaminar outras máquinas, inclusive IPs que não estão alocados, e não estão na tabela completa. Se houver rota *default*, o seu roteador vai encaminhar esse tráfego não válido para frente. Essa é uma das razões de se utilizar DFZ. Uma sugestão para solucionar esse problema é criar uma rota *default* e apontar para Null0 ou DevNull. Além disso, deve-se desabilitar o envio das mensagens 'ICMP unreachable', porque quando um roteador descarta um pacote, ele envia uma mensagem 'ICMP unreachable' avisando, porém, se o destino não é válido, não há necessidade de avisar a origem, isso apenas consome CPU desnecessariamente.

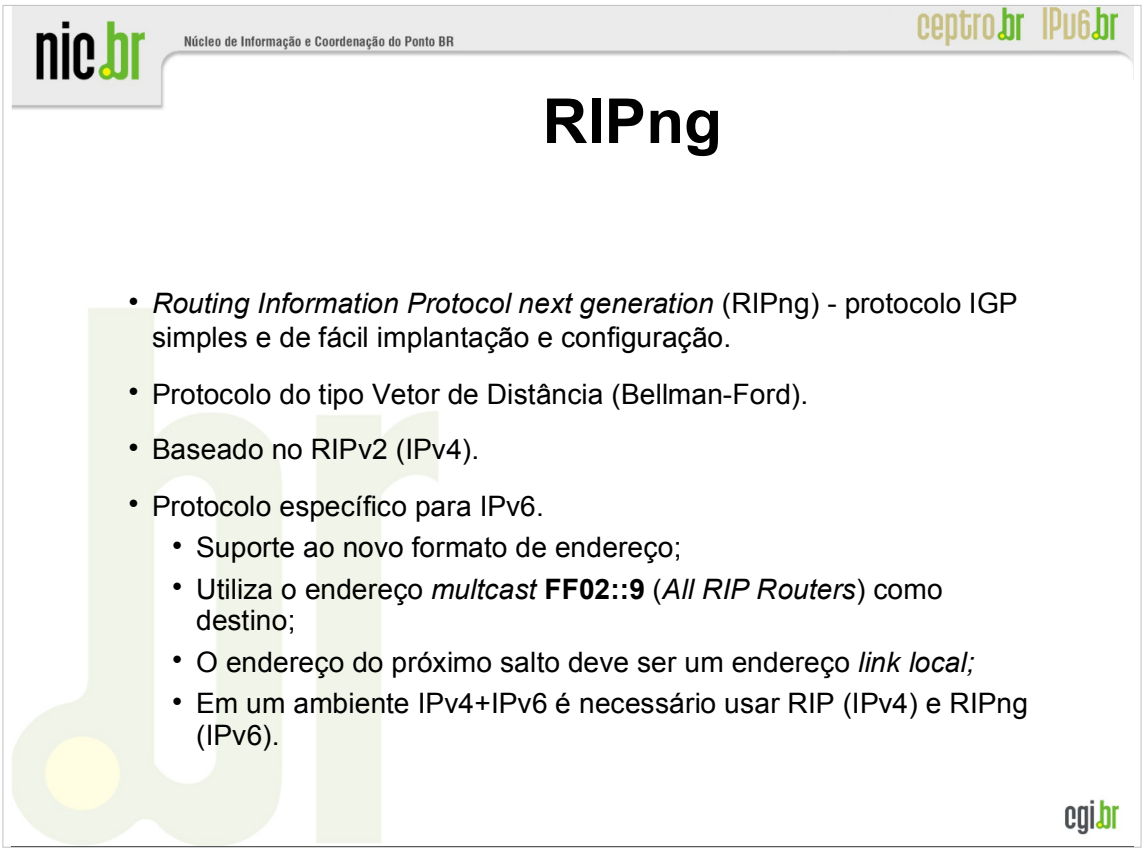
A rota *default* em IPv4 é 0.0.0.0/0 e em IPv6 ::/0.



Hoje há duas principais opções para se trabalhar com roteamento interno, o OSPF e o IS-IS. Esses dois protocolos são do tipo *Link-State*, isto é consideram as informações de estado do enlace, e mandam atualizações de forma otimizada, apenas quando há mudança de estado. Eles também permitem que se trabalhe com estrutura hierárquica, separando a rede por regiões. Isso é um ponto fundamental para IPv6.

Uma outra opção é o protocolo RIP (*Routing Information Protocol*). É um protocolo do tipo Vetor de Distância (Bellman-Ford), de fácil implementação e de funcionamento simples, porém apresenta algumas limitações como o fato de enviar sua tabela de estados periodicamente, independente de mudanças ou não na rede.

É importante que o protocolo de roteamento interno seja habilitado apenas nas interfaces onde são necessárias. Embora pareça óbvio, há quem configure de forma errada fazendo com que os roteadores fiquem tentando estabelecer vizinhança com outros ASs.



Para tratar o roteamento interno IPv6 foi definida uma nova versão do protocolo RIP, o *Routing Information Protocol next generation* (RIPng). Esta versão foi baseada no RIPv2 utilizado em redes IPv4, porém, ela é específica para redes IPv6.

Como mudanças principais destaca-se:

- Suporte ao novo formato de endereço;
- Utiliza o endereço *multcast* FF02::9 (*All RIP Routers*) como destino;
- O endereço do próximo salto deve ser um endereço *link local*.

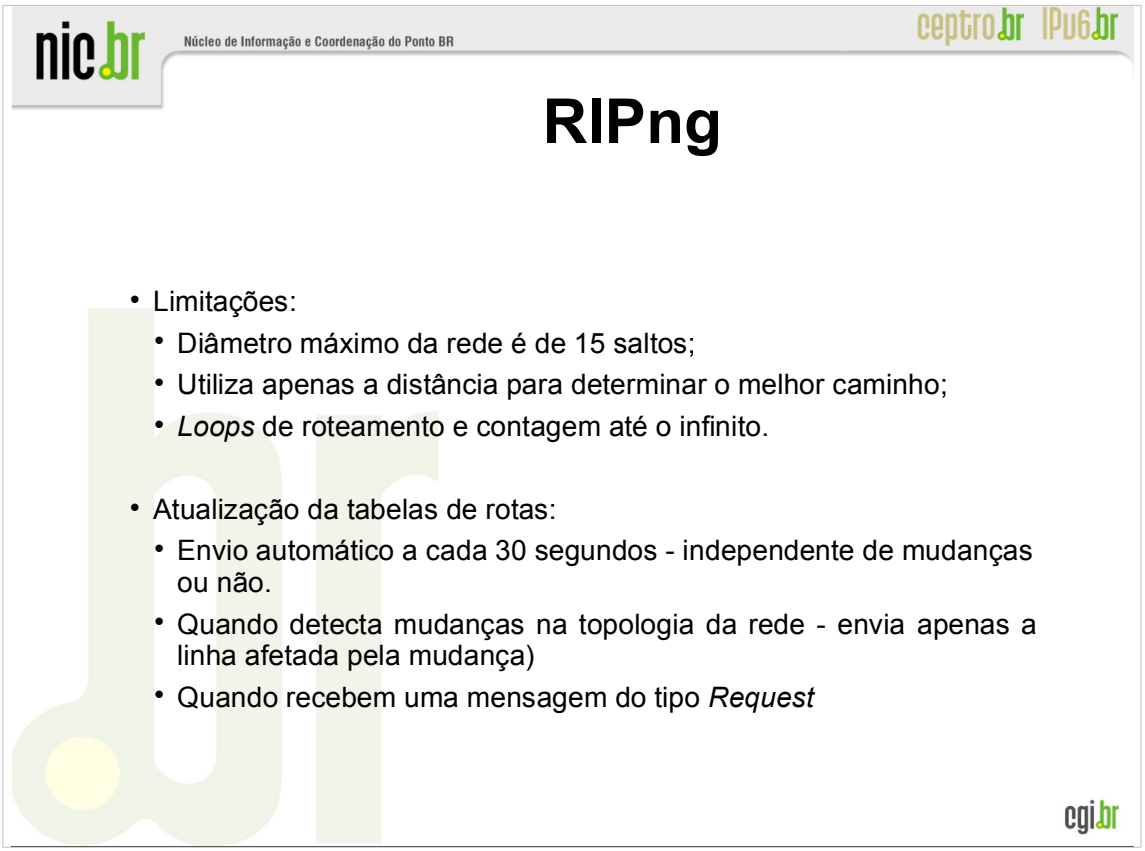
Em um ambiente com Pilha Dupla (IPv4+IPv6) é necessário usar uma instância do RIP para IPv4 e uma do RIPng para o roteamento IPv6.

Apesar de ser um protocolo novo, o RIPng ainda apresenta as mesma limitações das versões anteriores utilizadas com IPv4, como:

- Diâmetro máximo da rede é de 15 saltos;
- Utiliza apenas a distância para determinar o melhor caminho;
- *Loops* de roteamento e contagem até o infinito.

Mais informações:

- RFC 2080 - *RIPng for IPv6*



As informações presentes na tabela de rotas são:

- Prefixo do destino
- Métrica
- Próximo salto
- Identificação da rota (*route tag*)
- Mudança de rota
- Tempo até a rota expirar (padrão 180 segundos)
- Tempo até a "coleta de lixo" (*garbage collection*) (padrão 120 segundos)

A atualização da tabelas de rotas pode ocorrer de três formas: através do envio automático dos dados a cada 30 segundos; quando é detectada alguma mudanças na topologia da rede, enviando apenas a linha afetada pela mudança; e quando é recebida uma mensagem do tipo *Request.*

nic.br Núcleo de Informação e Coordenação do Ponto BR — ceptro.br IPv6.br

RIPng

- Mensagens *Request* e *Response*

8 bits	8 bits	16 bits
Comando	Versão	Reservado
Entrada 1 da tabela de rotas (RTE)		
....		
Entrada n da tabela de rotas		

- RTE
  - Prefixo IPv6 (128 bits)
  - Identificação da rota (16 bits)
  - Tamanho do prefixo (8 bits)
  - Métrica (8 bits)
- Diferente do RIPv2, o endereço do próximo salto aparece apenas uma vez, seguido de todas as entradas que devem utilizá-lo.

cgi.br

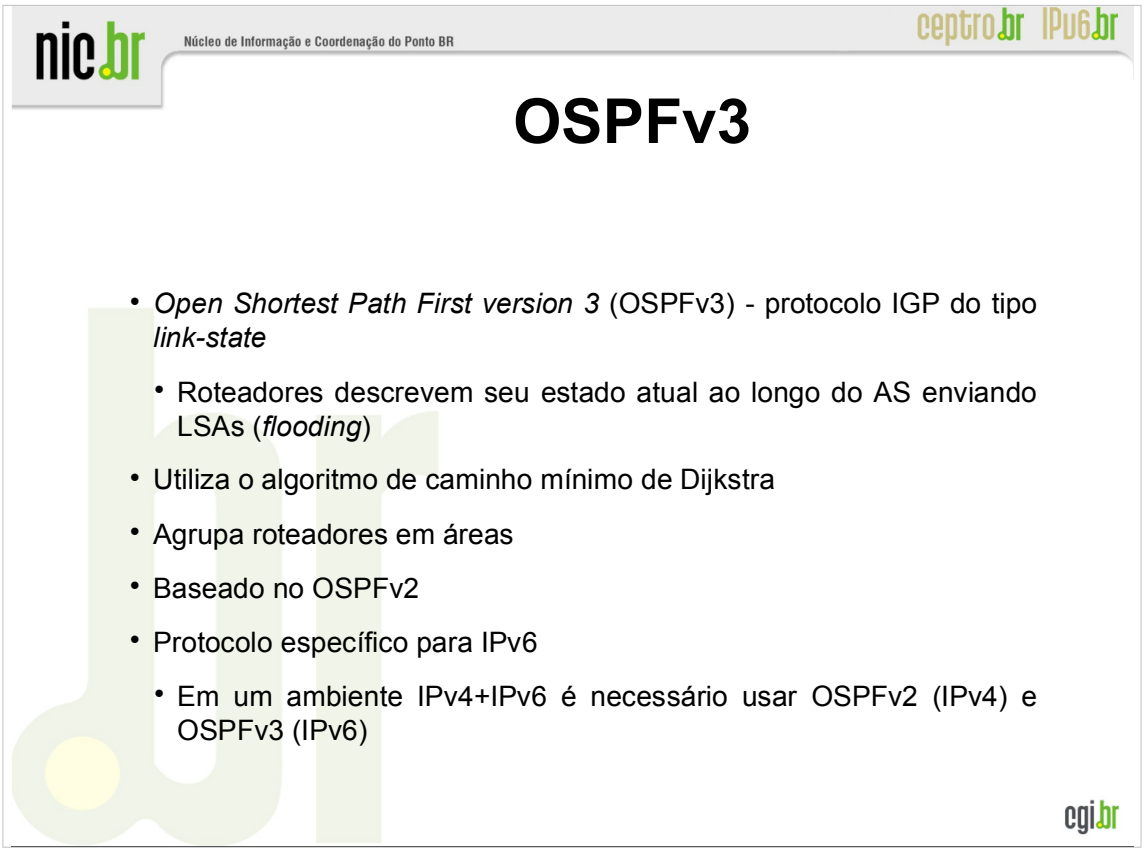
O cabeçalho das mensagens do RIPng é bem simples, composto pelos seguintes campos:

- Comando (*command*) – indica se a mensagem é do tipo *Requet* ou *Response*;
- Versão (*version*) – indica a versão do protocolo que atualmente é 1.

Esses campos são seguidos das entradas da tabela de rotas (*Route Table Entry* – RTE):

- Prefixo IPv6 (128 bits);
- Identificação da rota (16 bits);
- Tamanho do prefixo (8 bits);
- Métrica (8 bits).

Diferente do RIPv2, o endereço do próximo salto aparece apenas uma vez, seguido de todas as entradas que devem utilizá-lo

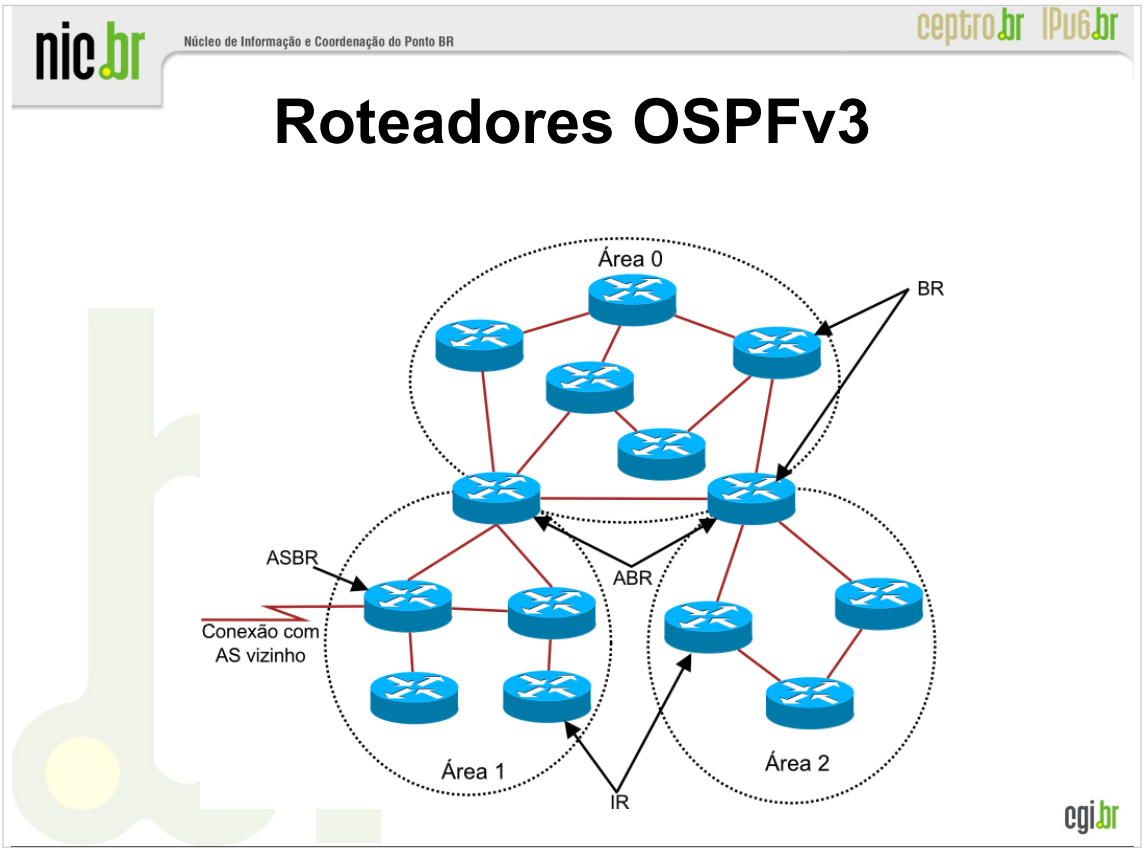


O OSPF é um protocolo do tipo *link-state* onde, através do processo de *flooding* (inundação), os roteadores enviam *Link State Advertisements* (LSA) descrevendo seu estado atual ao longo do AS. O *flooding* consiste no envio de um LSA por todas as interfaces de saída do roteador, de modo que todos os roteadores que receberem um LSA também o enviam por todas as suas interfaces. Com isso, o conjunto dos LSAs de todos os roteadores forma um banco de dados de estado do enlace, onde cada roteador participante do AS possui um banco de dados idêntico. Com as informações desse banco, o roteador, através do protocolo OSPF, constrói um mapa da rede que será utilizado para determinar uma árvore de caminhos mais curtos dentro de toda a sub-rede, tendo o próprio nó como raiz. Ele utiliza o algorítimo de Dijkstra para a escolha do melhor caminho e permite agrupar os roteadores em áreas.

O OSPF pode ser configurado para trabalhar de forma hierárquica, dividindo os roteadores de um AS em diversas áreas. A cada uma dessas áreas é atribuído um identificador único (Area-ID) de 32 bits e todos os roteadores de uma mesma área mantém um banco de dados de estado separado, de modo que a topologia de uma área é desconhecida fora dela, reduzindo a quantidade de tráfego de roteamento entre as partes do AS. A área de *backbone* é a responsável por distribuir as informações de roteamento entre as áreas *nonbackbone* e é identificada pelo ID 0 (ou 0.0.0.0). Em ASs onde não há essas divisões a área de *backbone* geralmente é a única a ser configurada.

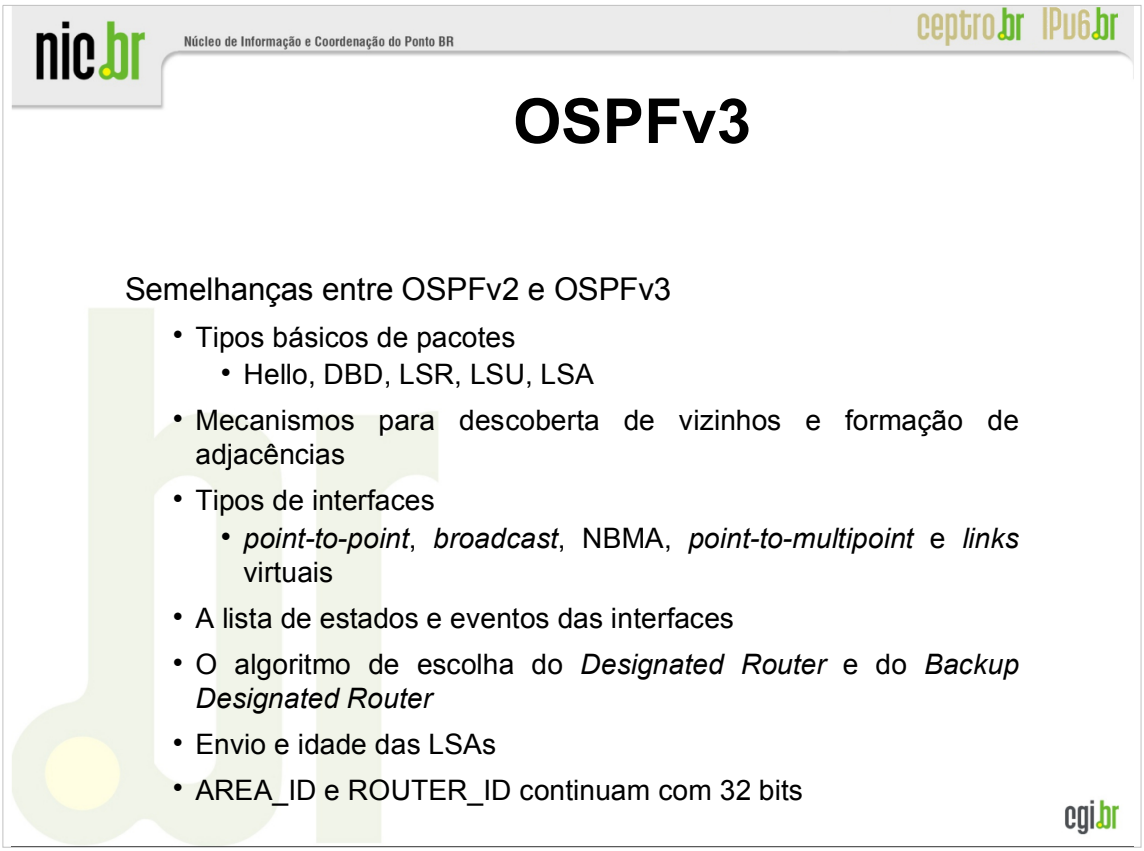
O OSPFv3 é um protocolo específico para IPv6, apesar de ter sido baseado na versão do OSPFv2, utilizada em redes IPv4. Deste modo, em uma rede com Pilha Dupla, é necessário utilizar OSPFv2 para o roteamento IPv4 e OSPFv3 para realizar o roteamento IPv6.

Mais informações:
- RFC 5340 - *OSPF for IPv6*



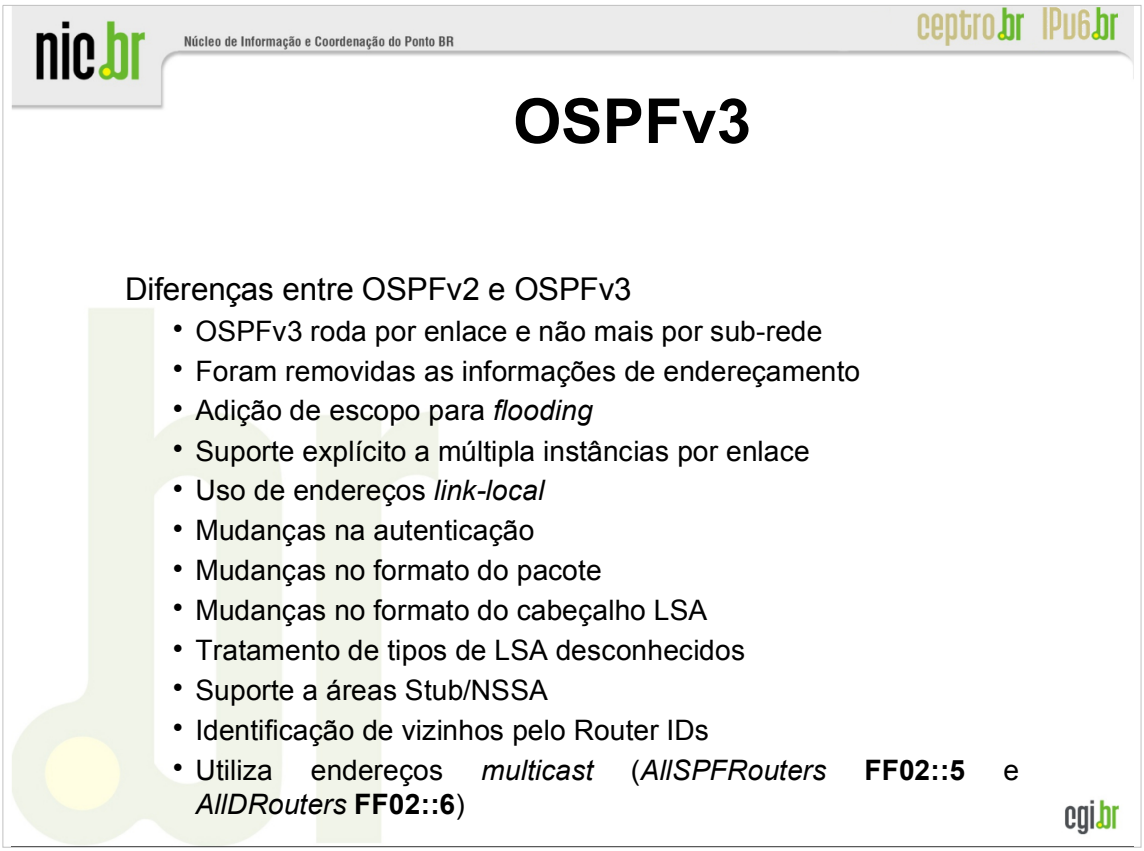
Os roteadores OSPF podem ser classificados como:

- *Internal Router* (IR) – roteadores que se relacionam apenas com vizinhos OSPF de uma mesma área;
- *Area Border Router* (ABR) – roteadores que conectam uma ou mais áreas ao *backbone*. Eles possuem múltiplas cópias dos bancos de dados de estado, uma para cada área, e são responsáveis por condensar as informações destas áreas e enviá-las ao *backbone*;
- *Backbone Router* (BR) – roteadores pertencentes a área *backbone*. Um ABR é sempre um BR, desde que todas suas áreas estejam diretamente conectadas ao *backbone* ou conectadas via *virtual link* - túnel que conecta uma área ao *backbone* passando através de outra área; e
- *Autonomous System Border Router* (ASBR) – roteadores que trocam informações de roteamento com roteadores de outro AS e distribuem as rotas recebidas ao longo do seu próprio AS.



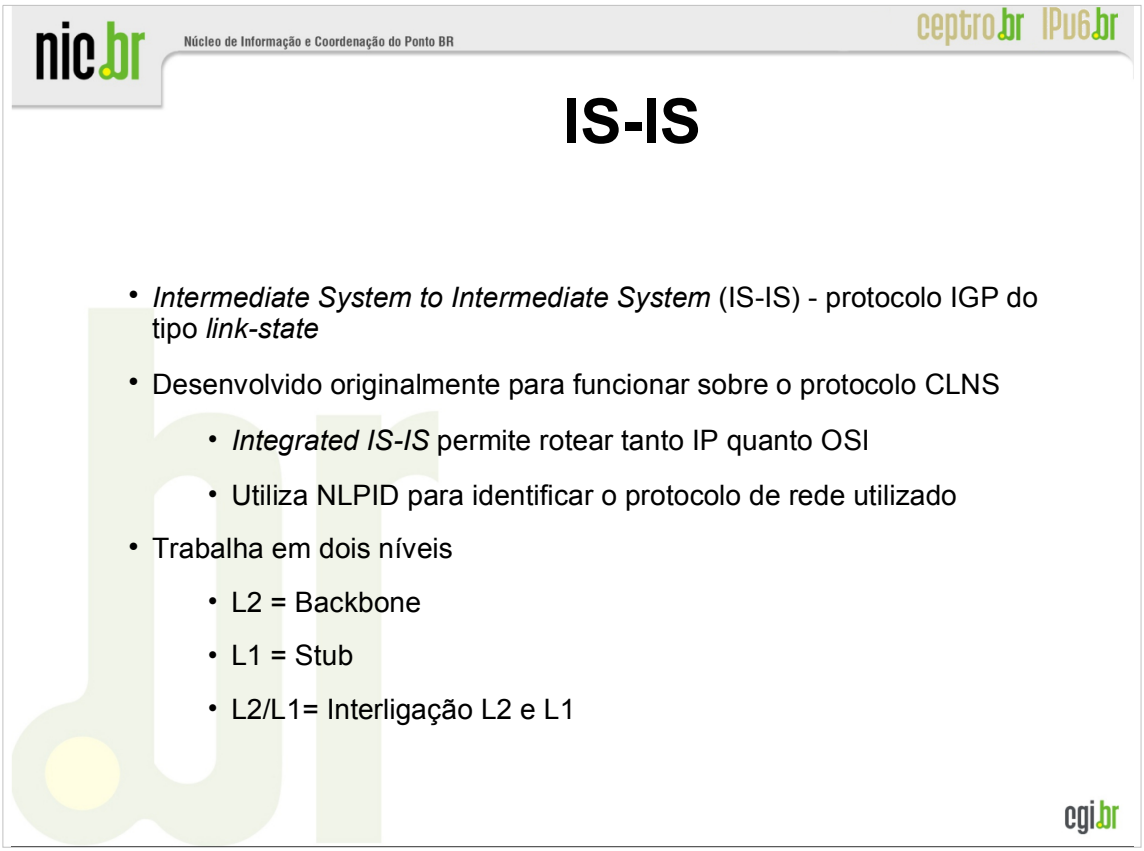
Algumas características do OSPFv2 ainda são encontradas no OSPFv3:

- Tipos básicos de pacotes
  - Hello, DBD, LSR, LSU, LSA
- Mecanismos para descoberta de vizinhos e formação de adjacências
- Tipos de interfaces
  - *point-to-point*, *broadcast*, NBMA, *point-to-multipoint* e *links* virtuais
- A lista de estados e eventos das interfaces
- O algoritmo de escolha do *Designated Router* e do *Backup Designated Router*
- Envio e idade das LSAs
- AREA_ID e ROUTER_ID continuam com 32 bits



Entras as principais diferenças entre o OSPFv2 e o OSPFv3 destacam-se:

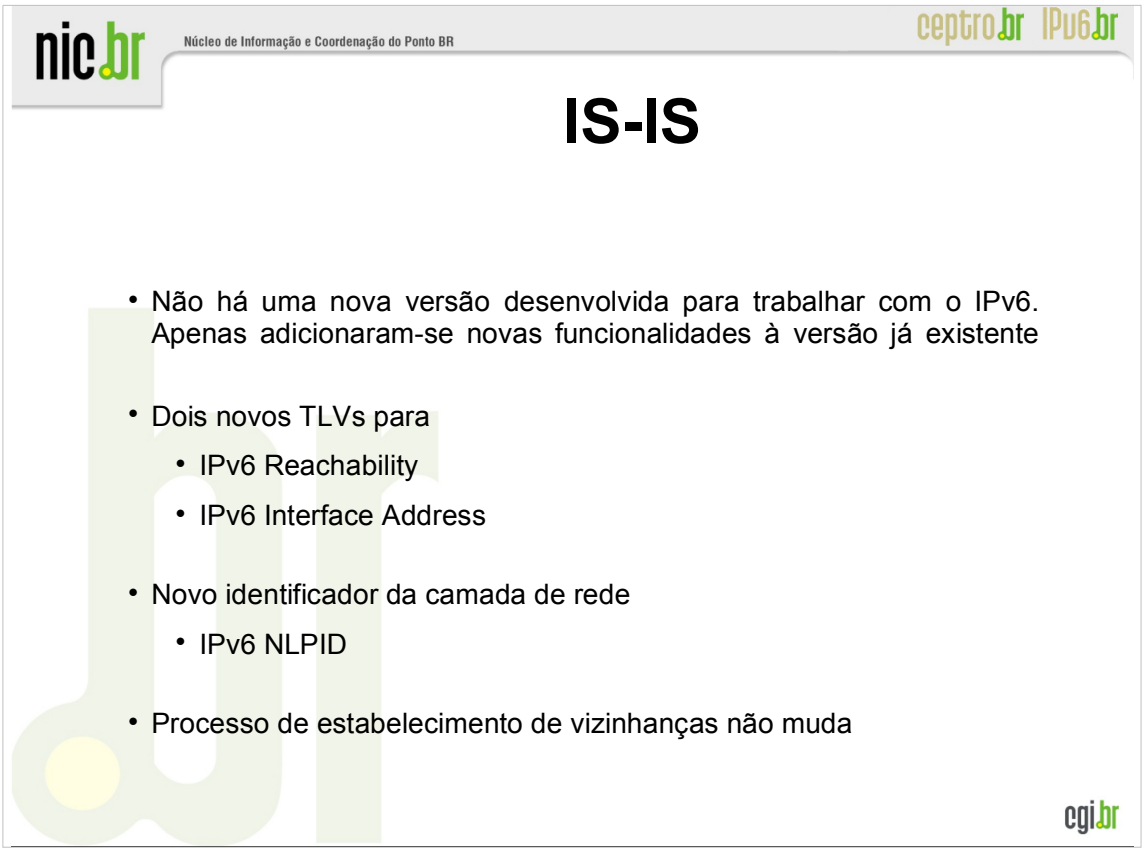
- OSPFv3 roda por enlace e não mais por sub-rede
- Foram removidas as informações de endereçamento
- Adição de escopo para *flooding*
- Suporte explícito a múltipla instâncias por enlace
- Uso de endereços *link-local*
- Mudanças na autenticação
- Mudanças no formato do pacote
- Mudanças no formato do cabeçalho LSA
- Tratamento de tipos de LSA desconhecidos
- Suporte a áreas Stub/NSSA
- Identificação de vizinhos pelo Router IDs
- Utiliza endereços *multicast* (*AllSPFRouters* **FF02::5** e *AllDRouters* **FF02::6**)



Assim como o OSFP, o *Intermediate System to Intermediate System* (IS-IS) é um protocolo IGP do tipo *link-state*, que utiliza o algorítimo de Dijkistra para calcular as rotas.

O IS-IS foi desenvolvido originalmente para funcionar sobre o protocolo CLNS, mas a versão *Integrated IS-IS* permite rotear tanto pacotes de rede IP quanto OSI. Para isso, utiliza-se um identificador de protocolo, o NLPID, para informar qual protocolo de rede está sendo utilizado.

Assim como o OSPF, o IS-IS também permite trabalhar a rede de forma hierárquica, atuando com os roteadores em dois níveis, o L1 (Stub) e o L2 (Backbone), além de roteadores que integram essas áreas, os L2/L1.



Para tratar o roteamento IPv6, não foi definida uma nova versão do protocolo IS-IS, apenas foram adicionadas novas funcionalidades à versão já existente.

Duas novas TLVs (*Type-Length-Values*) foram adicionadas:

- **IPv6 Reachability** (type 236) – carrega as informações das rede acessíveis;
- **IPv6 Interface Address** (type 232) – traz os endereços IP da interface que está transmitindo o pacote.

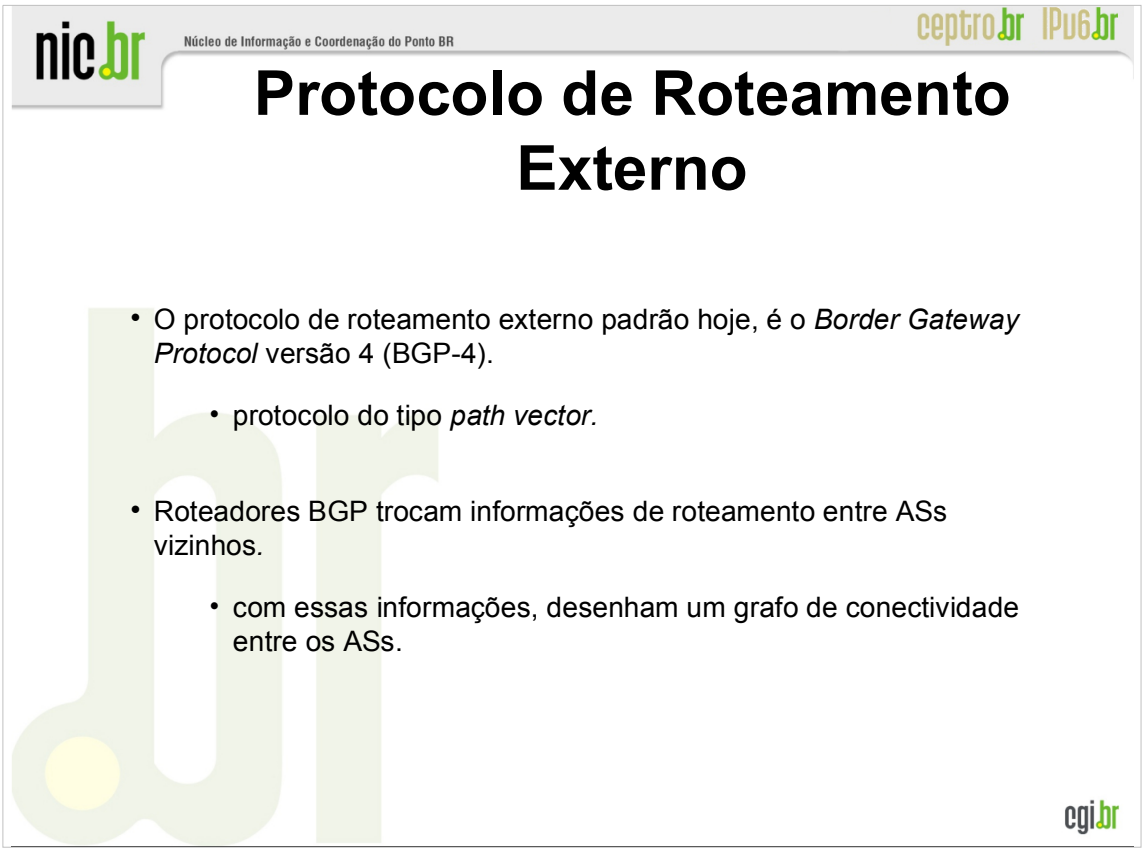
Também foi adicionado um novo identificador da Camada de Rede

- **IPv6 NLPID** – seu valor é 142.

O processo de estabelecimento de vizinhanças não muda.

Mais informações:

- RFC 1195 - *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*
- RFC 5308 - *Routing IPv6 with IS-IS*



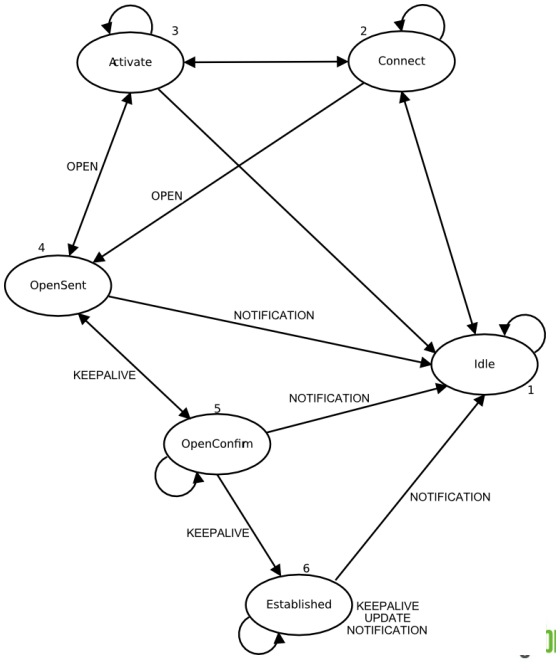
O protocolo de roteamento externo padrão hoje, é o *Border Gateway Protocol* versão 4 (BGP-4). É um protocolo do tipo *path vector*, onde roteadores BGP trocam informações de roteamento entre ASs vizinhos desenhando um grafo de conectividade entre os ASs.

nic.br Núcleo de Informação e Coordenação do Ponto BR

ceptro.br IPv6.br

BGP

- Porta TCP 179
- Quatro tipos de mensagem:
  - *Open*
  - *Update*
  - *Keepalive*
  - *Notification*
- Dois tipos de conexão:
  - eBGP
  - iBGP
- Funcionamento representado por uma Máquina de Estados.



O BGP é um protocolo extremamente simples baseado em sessões TCP ouvindo na porta 179.

Quatro tipos de mensagens BGP são utilizadas para troca de informações e manter o estado da conexão TCP:

- *Open* - enviada pelos dois vizinhos logo após o estabelecimento da conexão TCP, ela carrega as informações necessárias para o estabelecimento da sessão BGP, como ASN, versão do BGP, etc;
- *Update* – usada para transferir as informações de roteamento entre os vizinhos BGP, que serão utilizadas para construir o grafo que descreve o relacionamento entre vários ASs;
- *Keepalive* – são enviadas frequentemente para evitar que a conexão TCP expire;
- *Notification* – é enviada quando um erro é detectado, fechando a conexão BGP imediatamente após o seu envio.

Você pode estabelecer dois tipos de conexão BGP:

- externa (eBGP) – conexão entre dois ASs vizinhos;
- interna (iBGP) – conexão entre roteadores dentro de um mesmo AS. O estabelecimento do iBGP é muito importante para se manter uma visão consistente das rotas externas em todos os roteadores de um AS.

O funcionamento do BGP pode ser representado por uma Máquina de Estados Finitos. Para quem não está familiarizado com o BGP, ao verificar que o estado de uma conexão está “Active” ou “Established”, pode ter a falsa impressão de que a conexão está “ativa” ou “estabelecida”, mas em geral, em BGP quando há “palavras” representando o estado, significa que a sessão BGP ainda não está ok. A sessão só estará efetivamente estabelecida quando for observada a quantidade de prefixos que se está recebendo do vizinho. Esse nomes representam estados intermediários da sessão BGP. Identificar esses estados ajuda na análise e resolução de problemas.

Mais informações:
RFC 4271 - *A Border Gateway Protocol 4* (BGP-4)
RFC 4760 - *Multiprotocol Extensions for* BGP-4

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

Atributos do BGP

- O critério de seleção entre diferentes atributos do BGP varia de implementação para implementação.
- Os atributos BGP são divididos em algumas categorias e sub-categorias.

ORIGIN	Bem-conhecido	Mandatório
AS_PATH	Bem-conhecido	Mandatório
NEXT_HOP	Bem-conhecido	Mandatório
MULTI_EXIT_DISC	Opcional	Não-transitivo
LOCAL_PREF	Bem-conhecido	Discricionário
ATOMIC_AGGREGATE	Bem-conhecido	Discricionário
AGGREGATOR	Opcional	Transitivo

cgi.br

Apesar da RFC do BGP recomendar alguns pontos, o critério de seleção entre diferentes atributos do BGP pode variar de implementação para implementação. No entanto, a maior parte das implementações segue os mesmos padrões.

Os atributos BGP podem ser divididos em duas grandes categorias:

- **Bem-Conhecidos** (*Well-know*) – são atributos definidos na especificação original do protocolo BGP. Eles se subdividem em outras duas categorias:
  - **Mandatórios** (*Mandatory*) - devem estar sempre presentes nas mensagens do tipo UPDATE e devem ser obrigatoriamente reconhecidos em todas as implementações do protocolo;
  - **Descricionário** (*Discretionary*) - não precisam estar obrigatoriamente presente em todas as mensagens UPDATE.
- **Opcionais** (*Optional*) - não são obrigatoriamente suportados por todas as implementações de BGP. Eles se subdividem em outras duas categorias:
  - **Transitivos** (Transitive) – devem ser repassados nas mensagens UPDATE;
  - **Não-Transitivo** (*Non-transitive*) – não deve ser repassado.

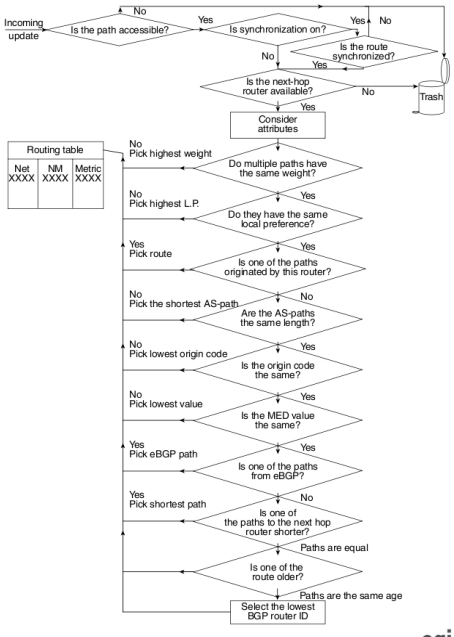
A RFC do BGP apresenta os seguintes atributos:

- *ORIGIN* – é Bem-Conhecido e Mandatório. Indica se o caminho foi aprendido via IGP ou EGP;
- *AS_PATH* - é Bem-Conhecido e Mandatório. Indica o caminho para se chegar a um destino, listando os ASN pelos quais se deve passar;
- *NEXT_HOP* – é Bem-Conhecido e Mandatório. Indica o endereço IP da interface do próximo roteador;
- *MULTI_EXIT_DISC* – é Opcional e Não-Transitivo. Indica para os vizinhos BGP externos qual o melhor caminho para uma determinada rota do próprio AS, influenciando-os, assim, em relação a qual caminho deve ser seguido no caso do AS possuir diversos pontos de entrada;
- *LOCAL_PREF* – é Bem-Conhecido e Discricionário. Indica um caminho preferencial de saída para uma determinada rota, destinada a uma rede externa ao AS;
- *ATOMIC_AGGREGATE* – é Bem-Conhecido e Discricionário. Indica se caminhos mais específicos foram agregados em menos específicos.
- *AGGREGATOR* - é Opcional e Transitivo. Indica o ASN do último roteador que formou uma rota agregada, seguido de seu próprio ASN e endereço IP.

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

Atributos do BGP

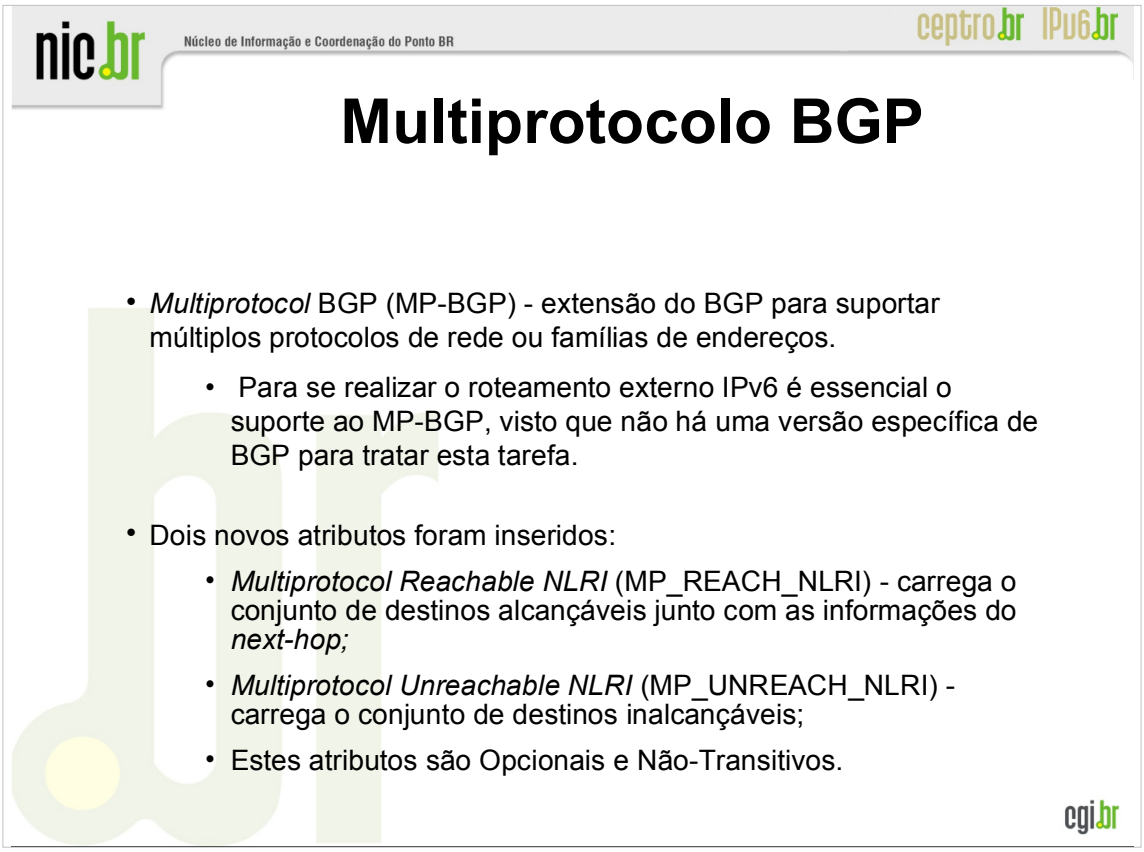
- Os atributos são considerados se o caminho for conhecido, se houver conectividade, se for acessível e se o *next hop* estiver disponível.
- A forma de seleção pode variar de acordo com a implementação.
- O *LOCAL_PREFERENCE* é um atributo extremamente poderoso para influenciar o tráfego de saída.
- O valor do *LOCAL_PREFERENCE* é valido para todo o AS.



Seleção do caminho no BGP (CISCO). cgi.br

Na decisão pela melhor rota, os atributos são considerados se o caminho for conhecido, se houver conectividade, se for acessível e se o *next hop* estiver disponível. Porém, a forma de seleção pode variar de acordo com a implementação.

Um atributo que merece destaque é o *LOCAL_PREFERENCE*. Ele é um atributo extremamente poderoso para influenciar o tráfego de saída. Seu valor é valido para todo o AS, sendo repassado apenas nas sessões iBGP.



Foram definidas extensões para o BGP-4 com o intuito de habilitá-lo a carregar informações de roteamento de múltiplos protocolo da Camada de Rede (ex., IPv6, IPX, L3VPN, etc.). Para se realizar o roteamento externo IPv6 é essencial o suporte ao MP-BGP, visto que não há uma versão específica de BGP para tratar esta tarefa.

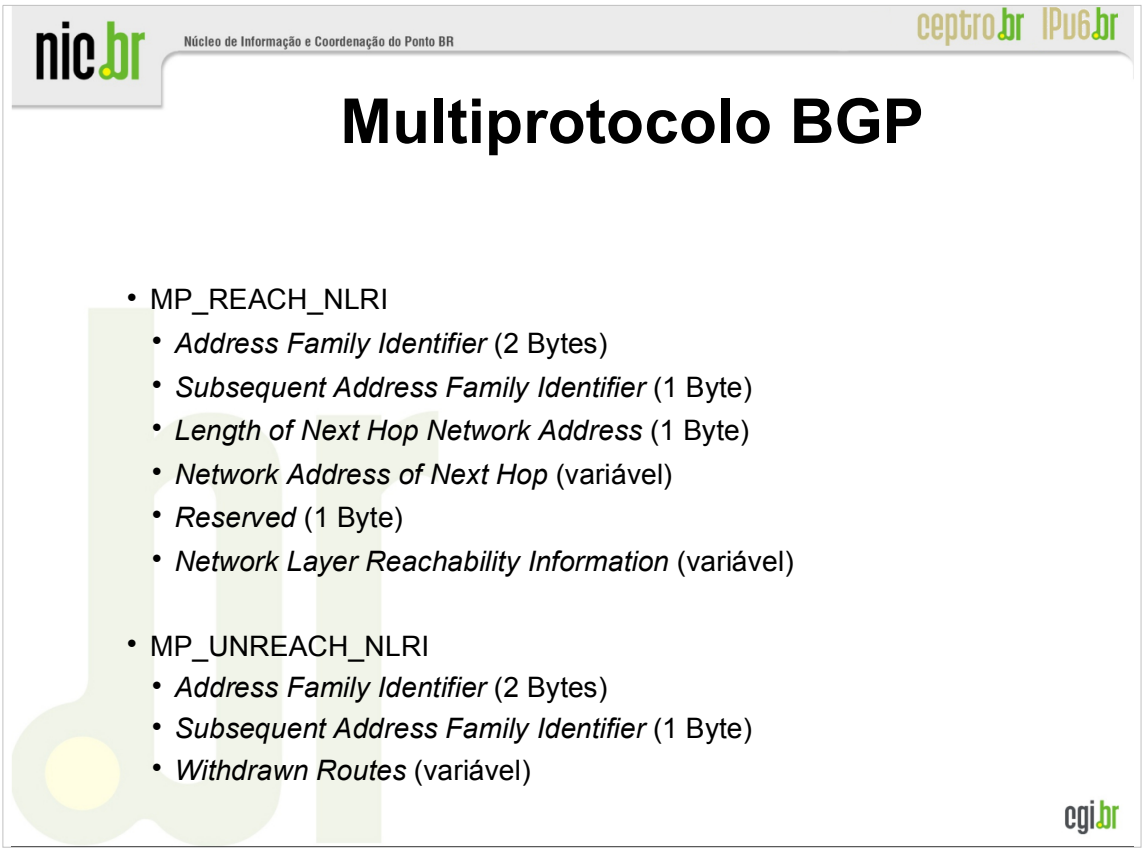
Para que o BGP possa trabalhar com informações de roteamento de diversos protocolos, dois novos atributos foram inseridos:

- *Multiprotocol Reachable NLRI* (MP_REACH_NLRI): carrega o conjunto de destinos alcançáveis junto com as informações do *next-hop;*
- *Multiprotocol Unreachable NLRI* (MP_UNREACH_NLRI): carrega o conjunto de destinos inalcançáveis.

Estes atributos são Opcionais e Não-Transitivos, e no caso de um roteador BGP não suportar MBGP, este deve ignorar estas informações, não passando-as para seus vizinhos.

Mais informações:

- RFC 2545 - *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*
- RFC 4760 - *Multiprotocol Extensions for BGP-4*



As seguintes informações são carregadas por esses atributos:

MP_REACH_NLRI

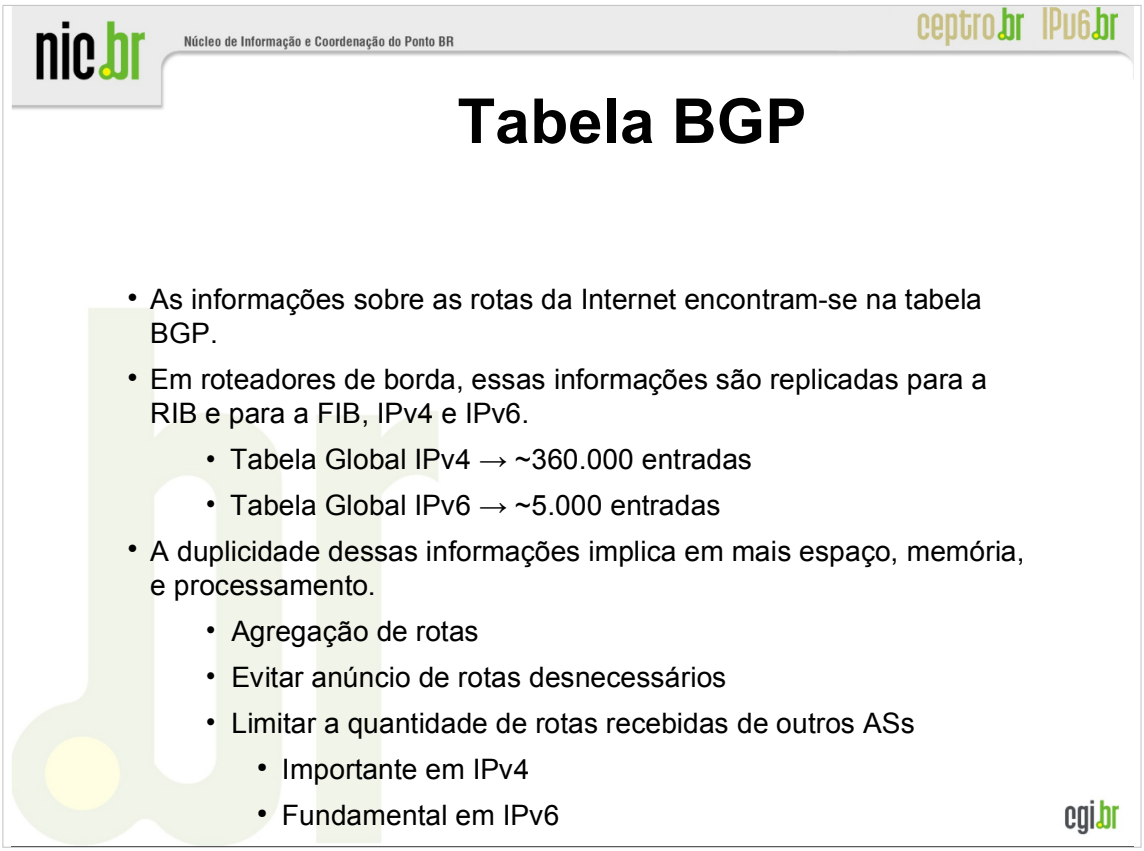
- *Address Family Identifier* (2 Bytes) - identifica o protocolo de rede a ser suportado;
- *Subsequent Address Family Identifier* (1 Byte) - identifica o protocolo de rede a ser suportado;
- *Length of Next Hop Network Address* (1 Byte) - valor que expressa o comprimento do campo *Network Address of Next Hop*, medida em Bytes;
- *Network Address of Next Hop* (variável) - contem o endereço do próximo salto;
- *Reserved* (1 Byte) - reservado;
- *Network Layer Reachability Information* (variável) - lista as informações das rotas acessíveis.

MP_UNREACH_NLRI

- *Address Family Identifier* (2 Bytes) - Identifica o protocolo de rede a ser suportado;
- *Subsequent Address Family Identifier* (1 Byte) - Identifica o protocolo de rede a ser suportado;
- *Withdrawn Routes* (variável) - lista as informações das rotas inacessíveis.

Códigos mais comuns para AFI e Sub-AFI

Código AFI	Código Sub-AFI	Significado
1	1	IPv4 Unicast
1	2	IPv4 Multicast
1	3	IPv4 based VPN
2	1	IPv6 Unicast
2	2	IPv6 Unicast e IPv6 Multicast RPF
2	3	Multicast RPF
2	4	IPv6 Label
2	128	IPv6 VPN
.....	....	...



As informações sobre as rotas da Internet encontram-se na tabela BGP. Em roteadores de borda, que tratam da comunicação entre ASs, essas informações são replicadas para a RIB e para a FIB, IPv4 e IPv6.

A tabela global IPv4 possuiu hoje aproximadamente 410.000 entradas. A tabela IPv6 possui aproximadamente 8.000 entradas. A duplicidade dessas informações em arquiteturas distribuídas, implica na necessidade de mais espaço para armazenamento, mais memória e mais processamento, tanto no módulo central quanto nas placas das interfaces.

Este dados implicam em outro aspecto importante, a necessidade de se estabelecer um plano hierárquico de endereçamento para minimizar a tabela de rotas e otimizar o roteamento, evitando o anúncio de rotas desnecessárias e desagregadas.

Os ASs também podem controlar os anúncios recebidos de seus vizinhos BGP. É possível, por exemplo, limitar o tamanho dos prefixos recebidos entre /20 e /24 IPv4, e entre /32 e /48 IPv6. Porém, lembre-se que podemos anunciar até 31 prefixos IPv4 (considerando anúncios entre um /20 e um /24) e 131.071 prefixos IPv6 (considerando anúncios entre um /32 e um /48), com isso, há quem controle também a quantidade de prefixos recebidos de seus vizinhos BGP, através de comandos como `maximum-prefix` (Cisco) e `maximum-prefixes` (Juniper). Tratar esta quentão em redes IPv4 é muito importante, mas em redes IPv6 é fundamental.

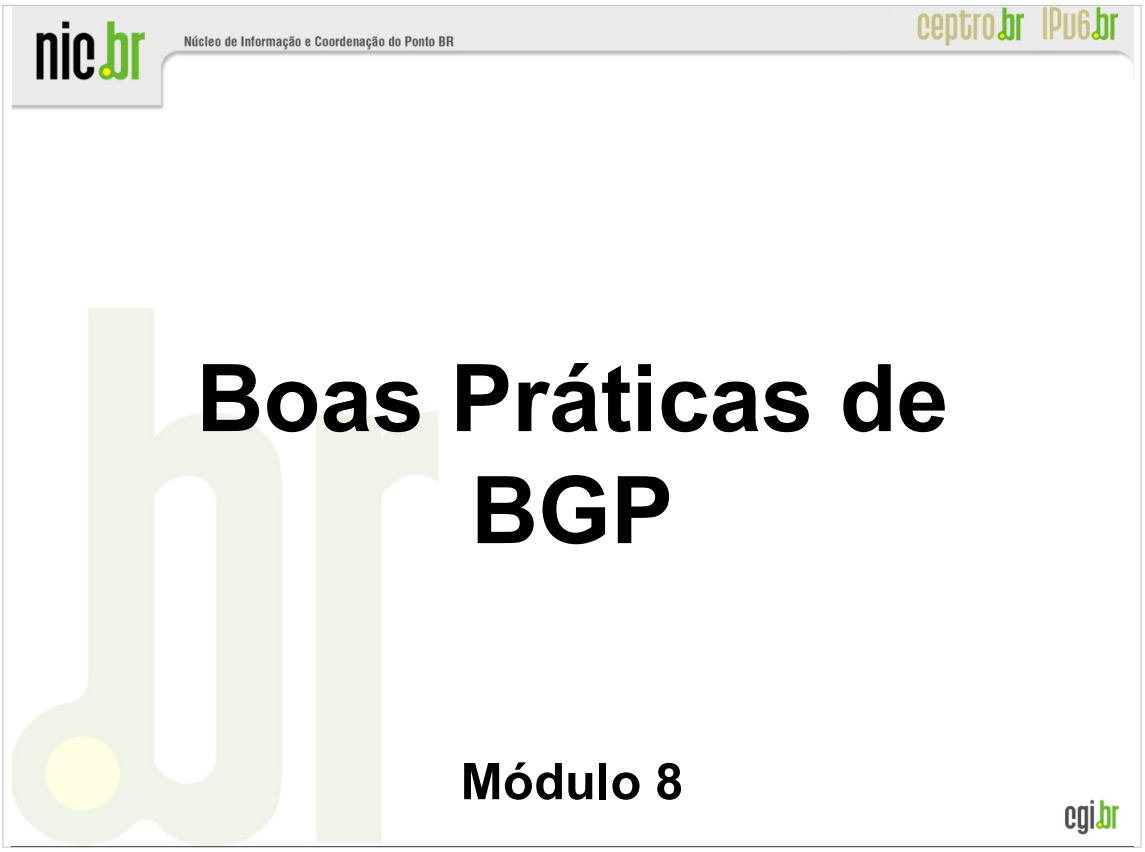
nic.br Núcleo de Informação e Coordenação do Ponto BR

ceptro.br IPv6.br

IPv6.br

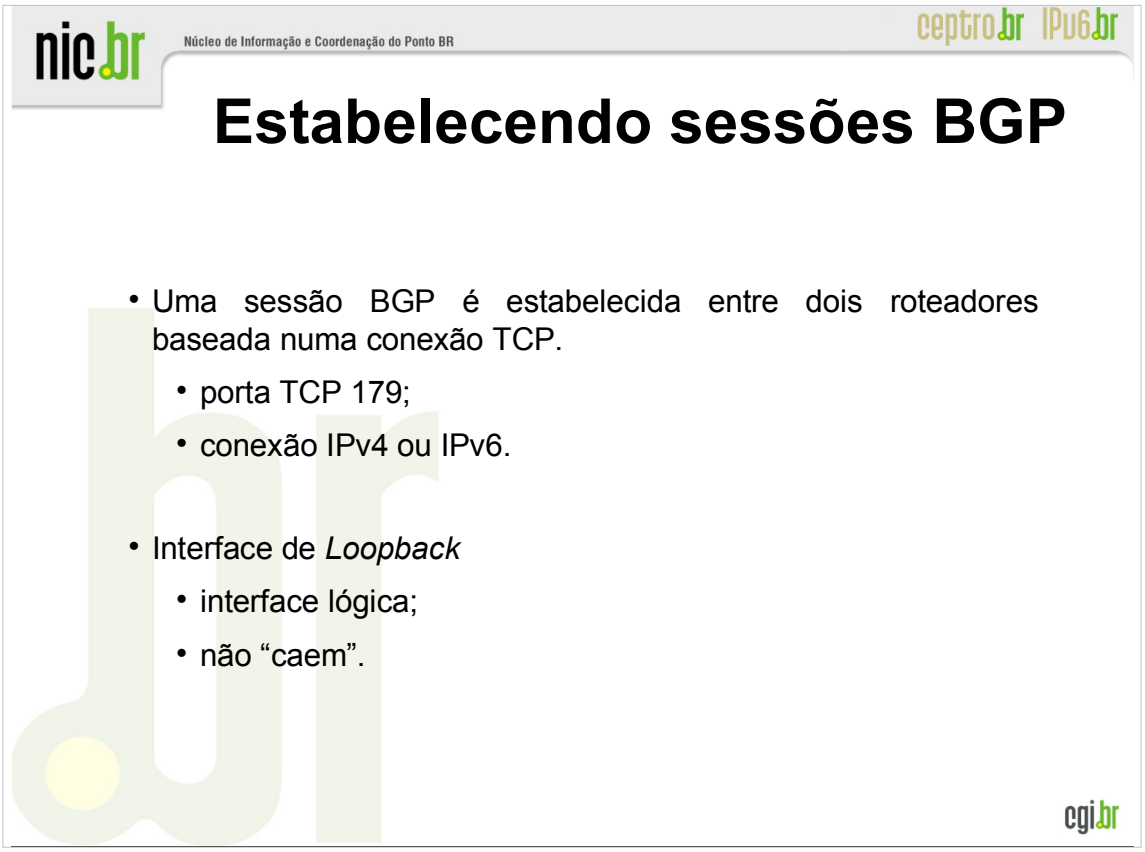
A Nova Geração do Protocolo Internet

cgi.br



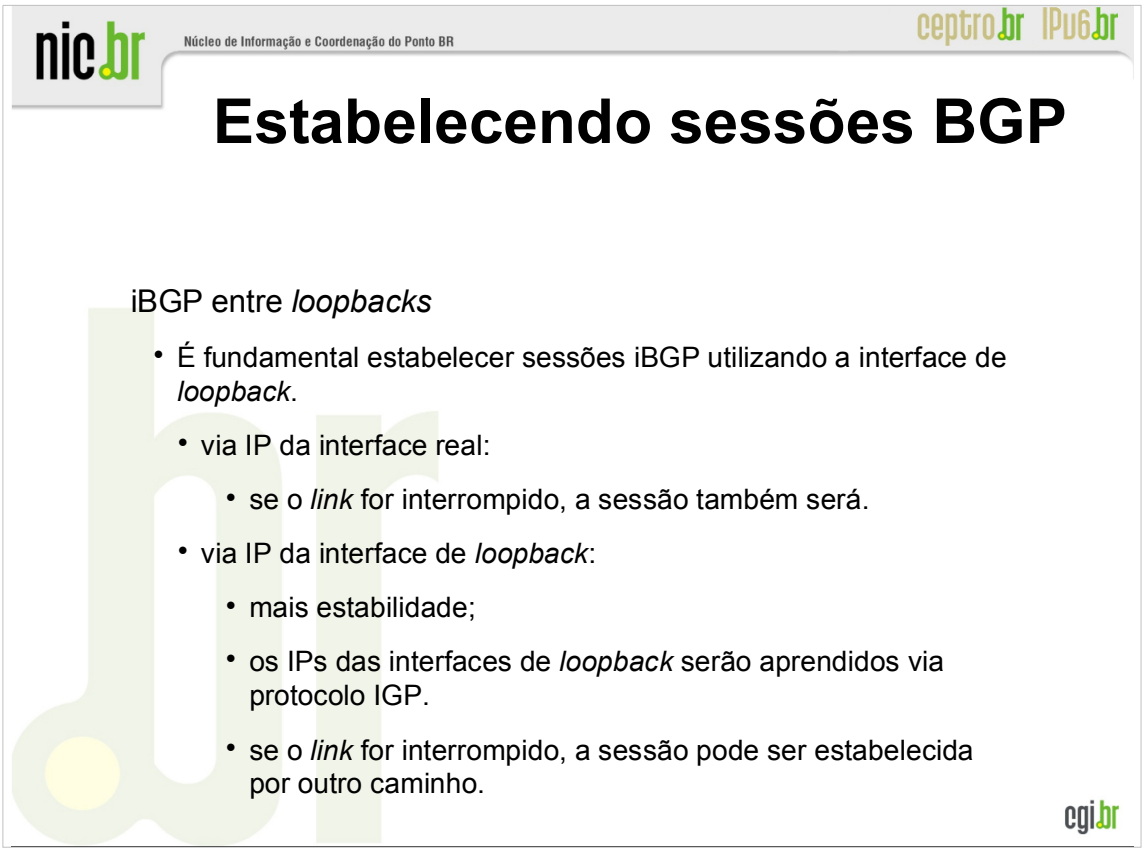
Neste módulo veremos alguns conceitos básicos de como as sessões BGP são estabelecidas; as vantagens na utilização de interfaces *loopbacks* em sessões iBGP e eBGP; alguns aspectos de segurança importantes que devem ser observados na comunicação entre ASs; formas de se garantir redundância e balanceamento de tráfego; além do detalhamento de uma série de comandos úteis para se verificar o estado das sessões BGP.

Todos esses tópicos serão abordados utilizando como base as plataformas Cisco, Quagga e Juniper, apresentando exemplos de configurações IPv6 e comparações a configurações IPv4.



Uma sessão BGP é estabelecida entre dois roteadores baseado-se em uma conexão TCP, utilizando como padrão a porta TCP 179, necessitando para isso, de uma conexão IP, seja IPv4 ou IPv6.

Uma das formas de se estabelecer essa comunicação, é através de interfaces *loopback*. Elas são interfaces lógicas, como a “null0”, ou seja, ela “não cai”, a não ser que se desligue o roteador, ou a interface seja desconfigurada.



Por suas características, é fundamental que as sessões iBGP sejam estabelecidas utilizando a interface *loopback*. Caso a sessão seja estabelecida via IP da interface física, real, se o *link* for interrompido, a sessão também será. Se for estabelecida via interface *loopback*, a sessão poderá ser re-estabelecida por outro caminho aprendido através dos protocolos IGP.

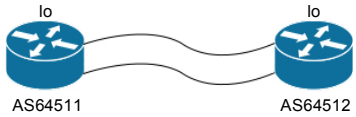
A utilização de interfaces *loopback* no estabelecimento das sessões iBGP proporcionam maior estabilidade aos AS.

nic.br Núcleo de Informação e Coordenação do Ponto BR

ceptro.br IPv6.br

Estabelecendo sessões BGP

eBGP entre *loopbacks*



- Balanceamento
  - Ex.:
    - Há dois roteadores e cada roteador representa um AS;
    - Eles estão conectados por dois *links;*
    - Utilizando o IP das interfaces reais:
      - Serão necessárias duas sessões BGP;
      - Eventualmente com políticas diferentes.
    - Utilizando o IP das interfaces de *loopbacks*
      - É estabelecida uma única sessão BGP;
      - Cria-se uma rota estática para o IP da interface *loopback* do vizinho através de cada *link.*

cgi.br

Também é recomendada a utilização de interfaces *loopback* no estabelecimento de sessões eBGP. Uma das finalidades de se estabelecer este tipo de conexão, é garantir balanceamento.

Analise o seguinte exemplo:

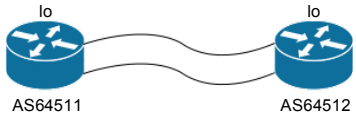
- Há dois roteadores, cada um representando um AS, e eles estão conectados através de dois *links*;
- Se for utilizado nesta comunicação o IP das interfaces reais será necessário estabelecer duas sessões BGP e para cada sessão, eventualmente haverá uma política diferente. Isso pode ocasionar complicações desnecessárias.
- Utilizar as interface *loopback* simplifica esse processo. Nesse caso, só será necessária uma sessão BGP, e a criação de rotas estáticas, apontando para o IP da *loopback* do vizinho através de cada *link*.

nic.br Núcleo de Informação e Coordenação do Ponto BR

ceptro.br IPv6.br

Estabelecendo sessões BGP

eBGP entre *loopbacks*



- Essa rota estática deve ser via interface ou via IP?
  - Se for uma interface serial pode-se apontar a rota para a interface;
  - Se for uma interface Ethernet deve-se apontar para o IP.
- Em *link* serial, o tamanho de rede IPv4 normalmente utilizado é /30.
  - Um /30 possui 4 IP; rede; *broadcast;* e os dois lados;
  - Em *links* seriais pode-se utilizar /31.
- Qual o equivalente ao /31 em IPv6?
- Em IPv6 pode-se trabalhar com redes /64 em *links* seriais.
- Uma boa opção é trabalhar com /112.

cgi.br

No estabelecimento da sessão eBGP através da *loopback*, a rota estática deve ser criada via interface ou via IP?

Se for uma interface serial pode-se apontar a rota para a interface. Interfaces seriais são um "tubo", as informações que trafegam por ela chegam diretamente do outro lado, portanto, pode-se apontar a rota para a interface.

Caso seja um meio compartilhado, uma interface Ethernet por exemplo, deve-se apontar para o IP.

Outro ponto importante é o tamanho de rede utilizado nestes tipos de *links*. Em um *link* serial, normalmente utiliza-se prefixos de redes /30 IPv4. Com isso, são possíveis quatro endereços IP: o de rede; de *broadcast*; e os dois que vão identificar as interfaces. No entanto, em um *link* serial não são necessários os endereços de rede nem de *broadcast*, por isso, há uma abordagem que utiliza prefixos de rede /31 neste tipo de *link*. Este método permite também a economia de uma grande quantidade de endereços IPv4, principalmente em operadora que trabalham com milhares de *links* ponto-a-ponto. Porém, isto só é recomendado para *links* seriais, não em Ethernet.

Em redes IPv6, o equivalente ao /31 IPv4 seria um /127. A RFC 3627 não recomenda a utilização de /127, devido a possíveis problemas com o endereço *anycast Subnet-Router*, no entanto, existe um *draft* que questiona esse argumento.

Existem diversas possibilidades para se trabalhar em IPv6. Em *links* ponto-a-ponto pode-se utilizar um prefixo /64 ou /126, mas uma opção interessante é utilizar /112, de modo a se trabalhar apenas com o último grupo de Bytes do endereço.

Mais informações:

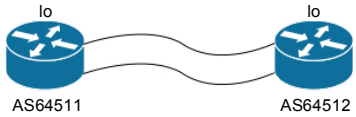
- RFC 3021 - *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*
- RFC 3627 - *Use of /127 Prefix Length Between Routers Considered Harmful*
- draft-kohno-ipv6-prefixlen-p2p-00.txt - *Use of /127 IPv6 Prefix Length on P2P Links Not Considered Harmful*

nic.br Núcleo de Informação e Coordenação do Ponto BR

ceptro.br IPv6.br

Estabelecendo sessões BGP

eBGP entre *loopbacks*



- Normalmente utilizam-se na interface de *loopback* prefixo /32 IPv4 ou /128 IPv6.
- O IP da *loopback* é de responsabilidade do próprio AS.
  - Não se deve utilizar IP privado.
- O IP do *link* de trânsito é da responsabilidade do Provedor de Trânsito.
  - Esse IP deve ou pode ser roteável?
    - Se for uma relação IP interna com a operadora, pode ser um IP válido, da operadora e não roteável, ex. conexão MPLS;
    - Se for um serviço Internet, o IP DEVE ser roteável.

cgi.br

Em relação ao endereçamento das interface *loopback*, normalmente utilizam-se prefixos /32 IPv4 e /128 IPv6. Isso desmistifica a ideia de que só é possível utilizar prefixos /64. A utilização de /64 só é obrigatória quando utiliza-se o protocolo de Descoberta de Vizinhança.

O endereço IP utilizado na *loopback* deve fazer parte do bloco do próprio AS e deve ser um endereço válido, não pode-se utilizar IPs privados. Os endereços IPs privados são para uso interno na sua rede e a comunicação entre ASs é uma conexão Internet, o que exige IPs válidos.

O endereço da interface real, conectada ao *link* de trânsito, deve ser do bloco da operadora que fornece o serviço, do *UpStream Provider*. Além disso, esse endereço IP deve ser roteável. Está é uma questão bastante polêmica, pois há quem defenda que esse endereço não seja roteável devido a questões de segurança. Se houver uma relação IP interna com a operadora, por exemplo uma conexão MPLS, é interessante que se utilize um IP válido, da operadora, e não roteável. No entanto, se for um serviço Internet, a operadora tem obrigação de fornecer um endereço roteável, porque ter conectividade é um ponto fundamenta em um serviço Internet.

Ex. 1 – O tráfego gerado por um roteador sai com IP da interface de saída como IP de origem. Com isso, se o seu roteador se conectar a um servidor NTP, seja IPv4 ou IPv6, o IP de origem desse pacote será o IP da interface pela qual ele sabe chegar ao destino. Se for um IP não roteável, o pacote vai chegar ao servidor, mas o servidor não saberá como retornar a resposta.

Ex. 2 - Há quem peça para a operadora não rotear esse IP por questões de segurança. Se os IPs internos forem roteáveis, não há proteção alguma, pois se houver apenas um IP roteável, ele será conhecido pelo mundo e haverá outro caminho para se chegar até ele, por exemplo, entrando em outro elemento do AS que conheça o IP do roteador.

nic.br Núcleo de Informação e Coordenação do Ponto BR

cetro.br IPv6.br

Estabelecendo sessões BGP

eBGP entre *loopbacks*

- Segurança
  - A utilização de interfaces *loopbacks* em sessões eBGP não é necessária apenas para garantir balanceamento.
  - Estabelecer sessões eBGP utilizando o IP da interface, facilita muito ataques contra a infraestrutura.
  - É recomendável trabalhar eBGP entre *loopbacks* mesmo que só haja um *link*.

cgi.br

Há quem defenda que a utilização de interface *loopback* só é realmente necessária apenas para garantir o balanceamento do tráfego. Essa prática auxilia também em relação a questões de segurança.

Estabelecer sessões eBGP utilizando o IP da interface, pode facilitar ataques contra a infraestrutura de um AS. Por isso, é recomendável trabalhar sessões eBGP entre *loopbacks* independente da existência de apenas um *link*.

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

Estabelecendo sessões BGP

eBGP entre *loopbacks*

- Segurança
  - Ex.:
    - Para estabelecer uma sessão eBGP sobre TCP são necessárias 4 informações básicas:
      - 2 IPs e 2 portas TCP (179 e >1024).
    - Se a sessão eBGP for estabelecida utilizando o IP da interface:
      - normalmente identifica-se um dos IP utilizando `traceroute`;
      - descobrindo o primeiro, descobre-se o segundo, visto que normalmente utiliza-se /30;
      - a terceira informação é uma porta padrão, a 179.
    - Ou seja, de 4 variáveis 3 podem ser descobertas de forma relativamente fácil.

cgi.br

Esta questão pode ser exemplificada com o cenário a seguir:

- Uma sessão eBGP entre dois roteadores é estabelecida através de uma conexão TCP;
- Para isso, são necessárias quatro variáveis básicas: dois endereços IP e duas portas TCP, uma padrão, a 179, e da mesma forma que uma aplicação HTTP, o roteador que iniciar a sessão BGP, vai sair por uma porta alta, maior que 1024, para fechar com a 179;
- Se a sessão eBGP for estabelecida utilizando o IP da interface, é possível identificar esse IP utilizando comandos como o *traceroute*. Como normalmente são utilizados prefixo /30 IPv4, se for descoberto um IP, descobrir o segundo torna-se mais simples. Com isso, das quatro variáveis, duas são fáceis de descobrir;
- A terceira é uma porta padrão, a 179;
- Ou seja de quatro variáveis três podem ser descobertas de forma relativamente fácil, e a única variável que falta, a porta alta, também não apresenta dificuldade para sua descoberta.

nic.br Núcleo de Informação e Coordenação do Ponto BR

cetro.br IPv6.br

Estabelecendo sessões BGP

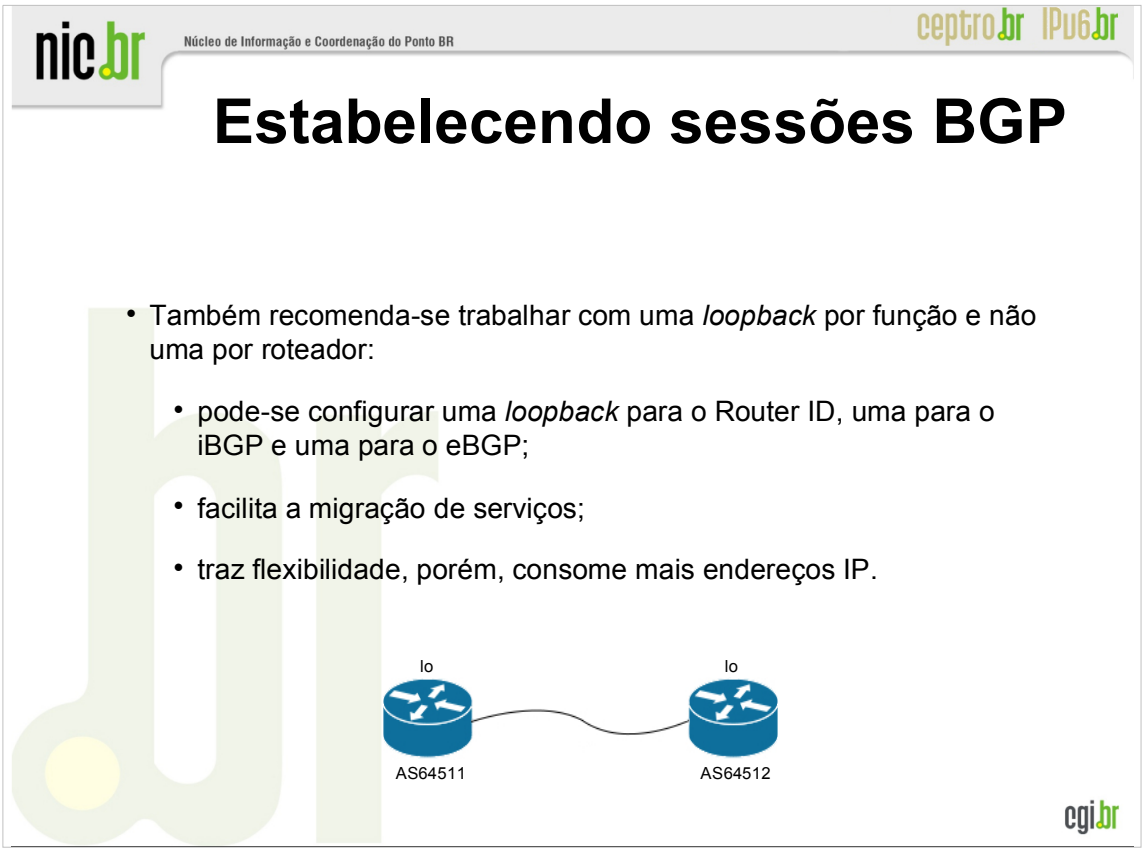
eBGP entre *loopbacks*

- Segurança
  - Uma das formas de derrubar um As ou um destino, é derrubar o AS que provê conectividade para ele.
  - Estabelecendo uma sessão eBGP utilizando *loopbacks:*
    - os IPs são das redes internas, não tendo relações entre eles;
    - dificulta a descoberta via `traceroute`.

cgi.br

Uma das formas de “derrubar” um AS ou um destino, é “derrubar” o AS que provê conectividade para ele e isso pode ser feito interrompendo as sessões eBGP.

Estabelecer a sessão eBGP utilizando a sessão *loopback* apresenta alguns pontos relativos a segurança. Os IPs da *loopback* são IPs da rede interna, não sendo descobertos com *traceroutes* facilmente, e o fato dos dois IPs serem totalmente distintos, não tendo relações entre eles, dificulta ainda mais.

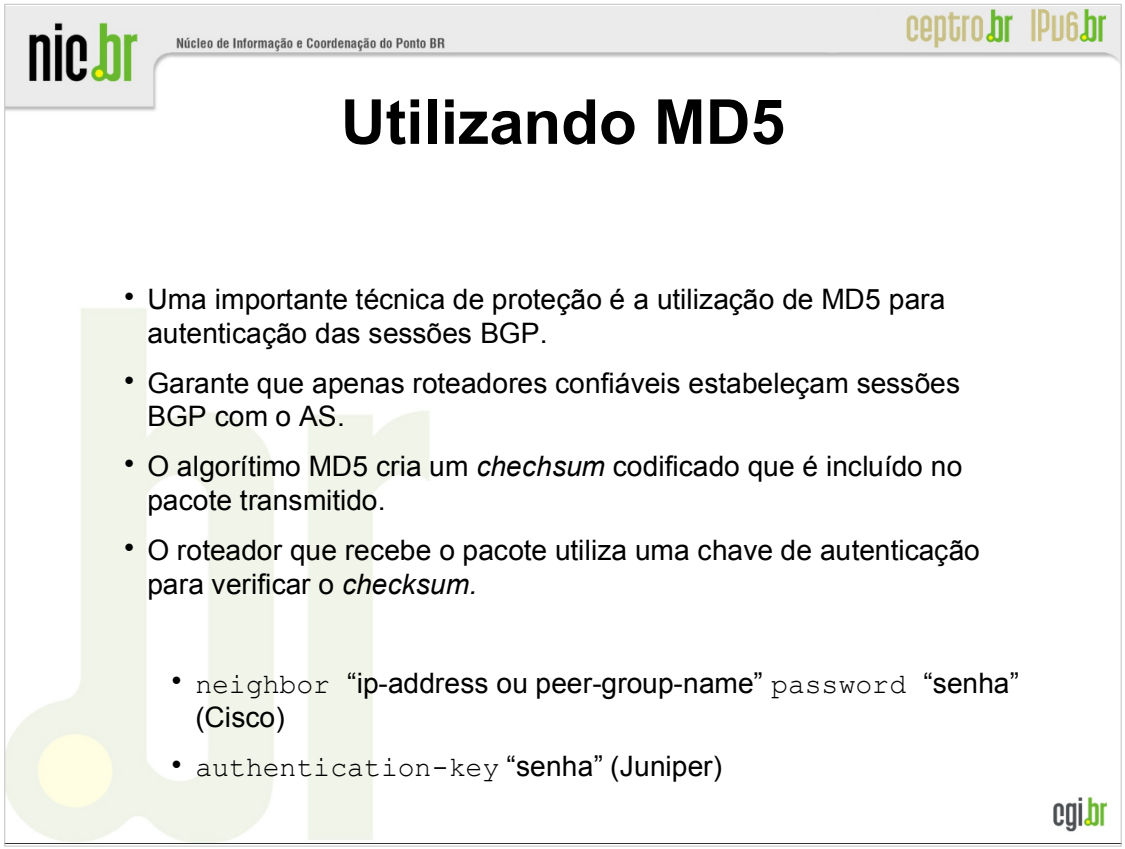


Outro aspecto que se deve destacar em relação à utilização da interface *loopback* é o de trabalhar com uma *loopback* por função e não uma única *loopback* por roteador. Por exemplo, pode-se configurar uma *loopback* para o Router ID, uma para o iBGP e uma para o eBGP.

Ex.:

- Há uma sessão eBGP estabelecida e é preciso migrar essa sessão para um outro roteador;
- Se além de ser a *loopback* usada para a sessão eBGP ela também for usada para *n* outras funções. Se for assim, a migração será mais complicada;
- Se for uma *loopback* por função, a migração poderá ser realizada sem interferir com as outras funções. Pode-se mudar o iBGP, o Router ID, alterar a sessão eBGP de um roteador para outro sem ter que avisar a operadora e não tem que mudar outros serviços internos.

Esta prática apresenta uma maior flexibilidade, apesar de consumir mais endereços IP. Entretanto, como normalmente são utilizados prefixos /32 ou /128, esta questão não é tão grave.



Uma importante técnica de proteção é a utilização de MD5 para autenticação das sessões BGP. Desta forma garante-se que apenas roteadores confiáveis estabeleçam sessões BGP com o AS.

O algorítimo MD5 cria um *chechsum* codificado que é incluído no pacote transmitido e o roteador que recebe o pacote utiliza uma chave de autenticação para verificar o *hash.*

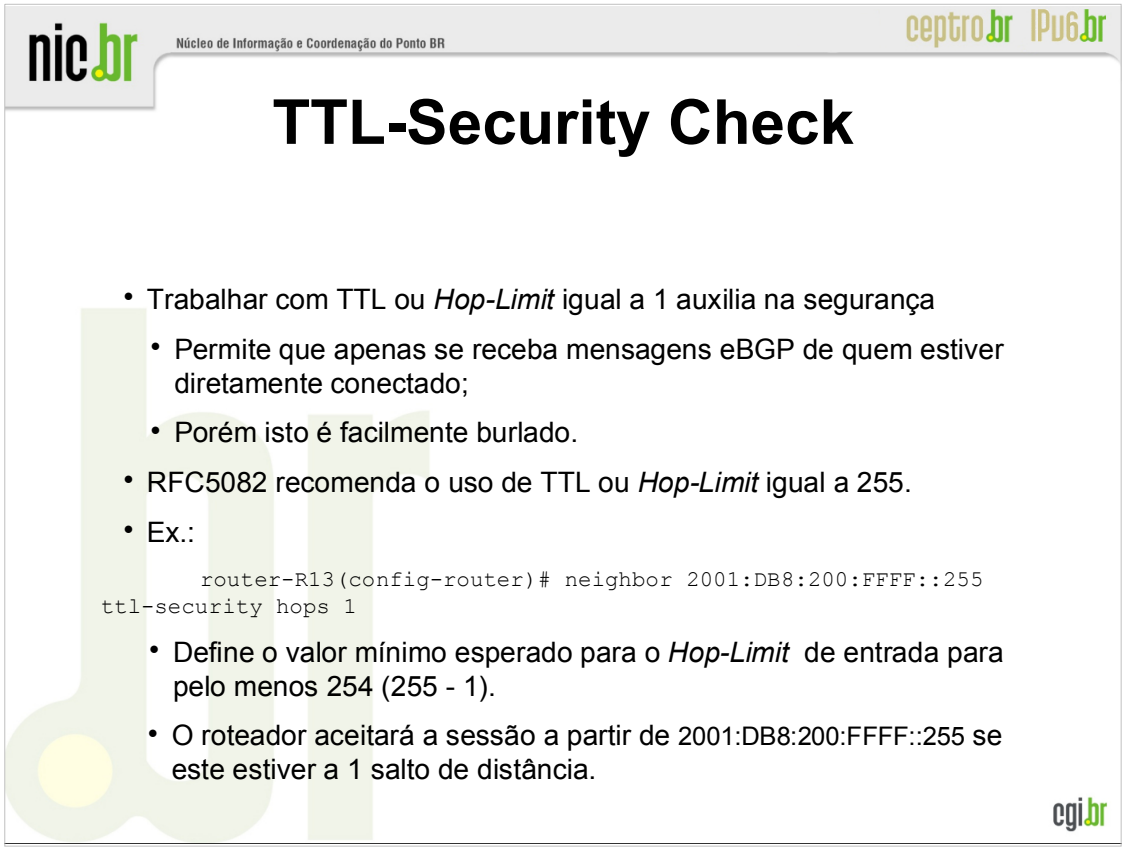
Utilize os seguintes comandos para habilitar essa funcionalidade:

`neighbor` "ip-address ou peer-group-name" `password` "senha" (Cisco)

`authentication-key` "senha" (Juniper)

Mais informações:

- RFC 1321 - *The MD5 Message-Digest Algorithm*
- RFC 2385 - *Protection of BGP Sessions via the TCP MD5 Signature Option*



Trabalhar com TTL ou *Hop-Limit* igual a 1 auxilia na segurança das sessões BGP, pois permite que apenas se receba mensagens eBGP de quem estiver diretamente conectado. Porém isso é facilmente burlado, basta utilizar comandos como `traceroute` para identificar quantos saltos são necessários para chegar ao roteador de destino e gerar um pacote como valor do TTL necessário para alcança-lo.

A RFC 5082 recomenda que se trabalhe com TTL ou *Hop-Limit* igual a 255 em vez de 1. Deste modo, em uma sessão eBGP diretamente conectada utilizando o IP da interface, é possível garantir que o vizinho BGP está a no máximo um salto de distância através da leitura do valor do TTL, que terá sido decrementado a cada hop.

Para utilizar essa funcionalidade é preciso configurar os dois vizinhos participantes da sessão eBGP. Quem envia a mensagem deve montar o pacote com TTL ou *Hop-Limit* igual a 255 e quem recebe deve habilitar a verificação desse campo. Em roteadores Cisco é possível fazer a verificação da seguinte forma:

```
router-R13(config-router)#neighbor 2001:DB8:200:FFFF::255 ttl-security hops 1
```

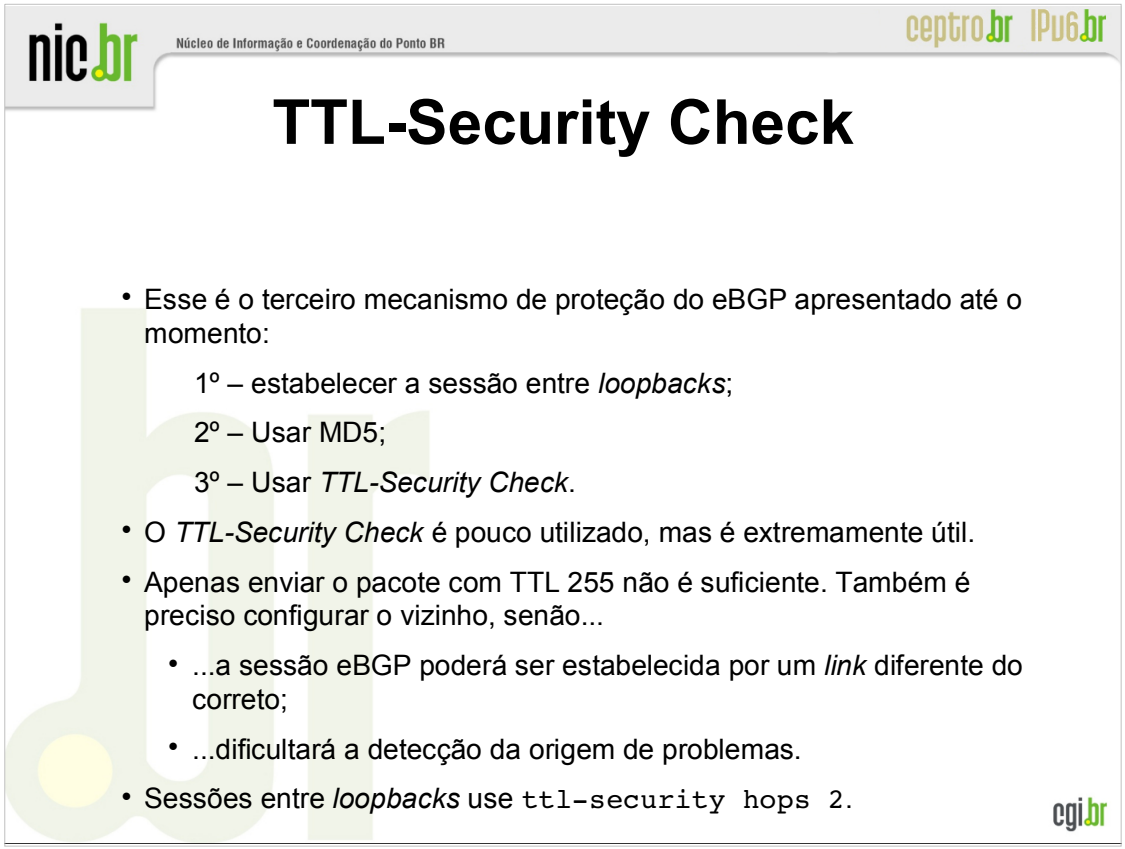
Deste modo, define-se o valor mínimo esperado para o *Hop-Limit* de entrada para pelo menos 254 (255 – 1). Com isso, o roteador aceitará a sessão a partir de `2001:DB8:200:FFFF::255` se este estiver a 1 salto de distância.

Com um vizinho BGP IPv4 essa linha seria:

```
router-R13(config-router)#neighbor 10.2.255.255 ttl-security hops 1
```

Mais informações:

- RFC 5082 - The Generalized TTL Security Mechanism (GTSM)
- http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_btsh.html
- http://www.juniper.net/us/en/community/junos/script-automation/library/configuration/ttl-security/



Esse é o terceiro mecanismo de proteção do eBGP apresentado até o momento:

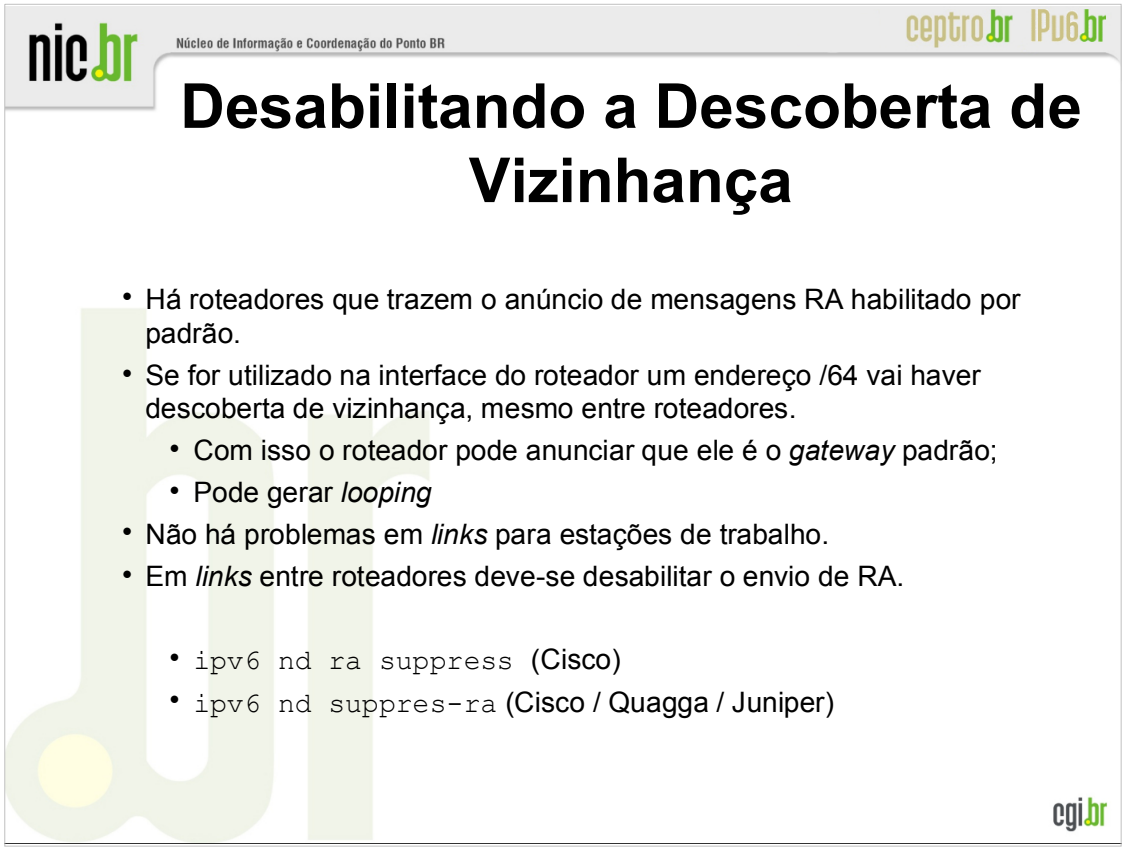
1º – estabelecer a sessão entre *loopbacks*;

2º – Usar MD5;

3º – Usar *TTL-Security Check*.

O *TTL-Security Check* é pouco utilizado, mas é extremamente útil. No entanto, apenas enviar o pacote com TTL 255 não é suficiente, também é preciso configurar o vizinho. Caso isso não ocorra, a sessão eBGP poderá ser estabelecida por um *link* diferente do correto, o que dificultará a detecção da origem de problemas.

Outro ponto importante, é que em sessões entre *loopbacks* deve-se usar `ttl-security hops 2`.



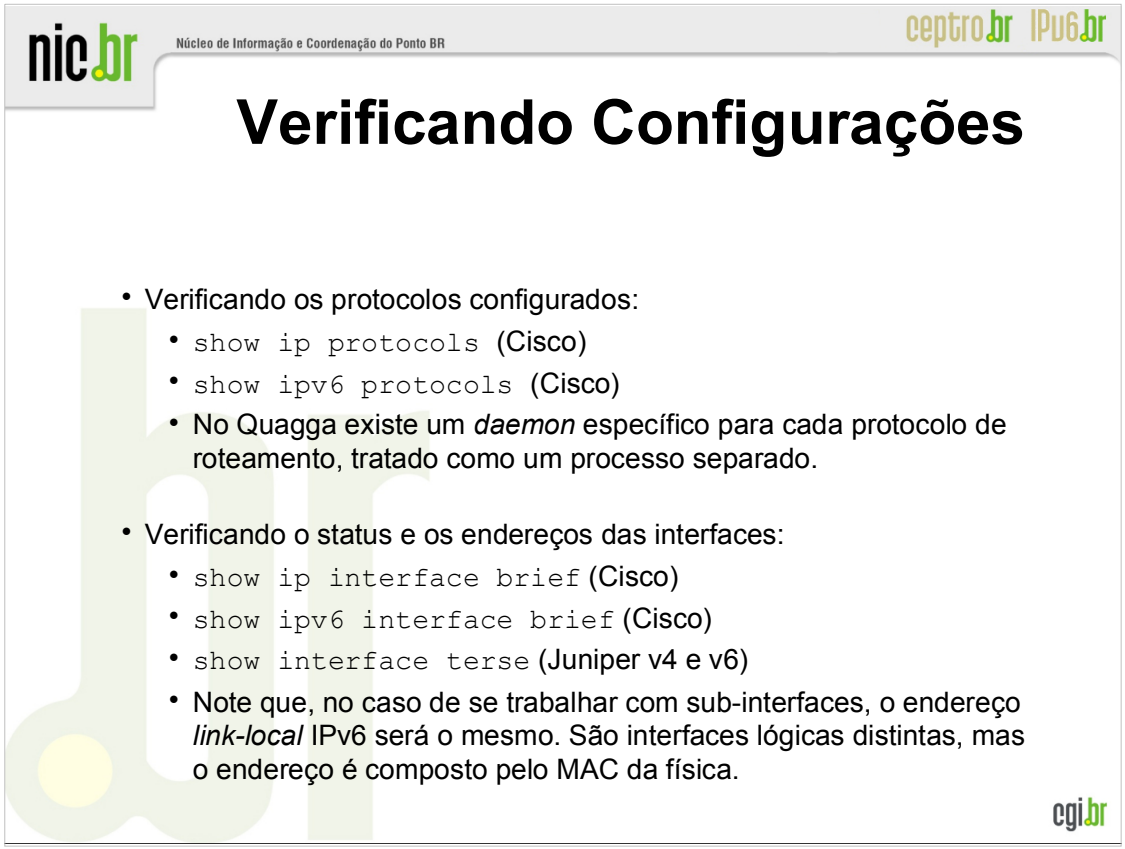
Um ponto importante na configuração de roteadores IPv6 é o funcionamento do protocolo de Descoberta de Vizinhança. Há roteadores que trazem o anúncio de mensagens RA (*Router Advertisements*) habilitado por padrão. Caso seja utilizado na interface do roteador um endereço / 64, vai haver descoberta de vizinhança, mesmo entre roteadores. Com isso o roteador pode anunciar que ele é o *gateway* padrão para os outros roteadores da rede, podendo gerar *loopings.*

Esta funcionalidade só deve ser habilitada em interfaces que estejam conectadas a estações de trabalho ou em alguns casos, a servidores. Em *links* entre roteadores deve-se desabilitar o envio das mensagens RA.

Esta funcionalidade pode ser desabilitada com a utilização dos seguintes comandos:

`ipv6 nd ra suppress` (Cisco)

`ipv6 nd suppres-ra` (Cisco / Quagga / Juniper)



Ter conhecimento das funcionalidades e configurações habilitas nos roteadores é muito importante principalmente quando se obtém um equipamento novo. Alguns comandos que podem facilitar essa tarefa são:

Para verificar os protocolos configurados podemos utilizar:

`show ip protocols` (Cisco)

`show ipv6 protocols` (Cisco)

No Quagga existe um *daemon* específico para cada protocolo de roteamento, tratado como um processo separado.

Para verificar o status e os endereços das interfaces utilizamos:

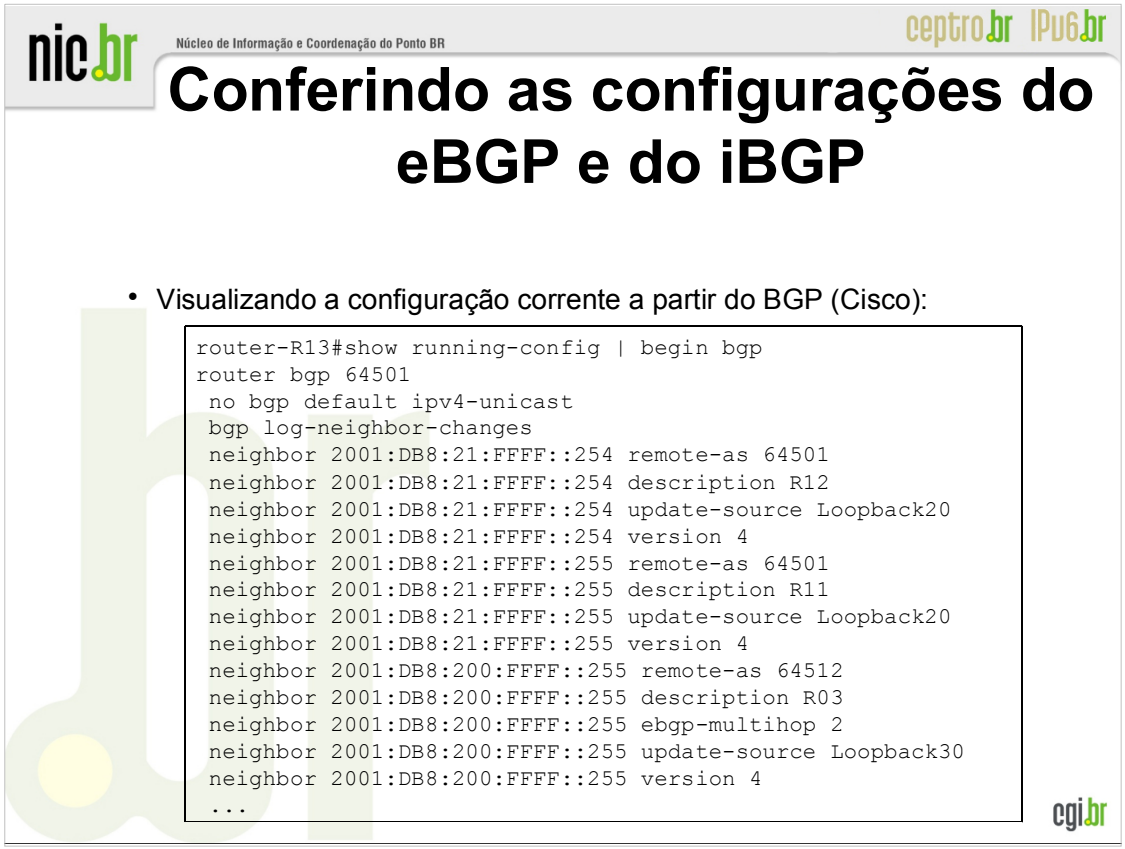
`show ip interface brief` (Cisco)

`show ipv6 interface brief` (Cisco)

`show interface terse` (Juniper v4 e v6)

Note que, no caso de se trabalhar com sub-interfaces, o endereço *link-local* IPv6 será o mesmo. São interfaces lógicas distintas, mas o endereço é composto pelo MAC da física.

Esses comandos são úteis porque mostram de forma resumida o está configurado no equipamento, que interfaces já estão habilitadas com IPv6 por exemplo, etc.



Existem alguns comando que podem facilitar a analisar e conferência das políticas de entrada e saída de um roteador.

Para visualiza a configuração corrente a partir do BGP em um roteador Cisco ou Quagga:

```
show running-config | begin bgp
```

Em nosso exemplo, a primeira linha da configuração BGP indica o ASN do próprio AS:

- `router bgp 64501`

Por padrão, os roteadores cisco e quagga só conhecem uma família de endereços, a *ipv4-unicast.* Para utilizar outras famílias de endereços, utiliza-se o conceito de *address-family*, e para habilitá-lo utiliza-se o comando:

- `no bgp default ipv4-unicast`

Para habilitar IPv6 em roteadores Cisco e Quagga recomenda-se que seja marcada uma janela de manutenção, para deste modo, poder interromper o tráfego, aplicando o comando

- `no router bgp 64501`

e refazer toda a configuração com *address-family.*

Mesmo utilizando *address-family*, no início da configuração sempre são apresentadas as informações gerais que independem da família. Por exemplo:

- `neighbor 2001:DB8:200:FFFF::255 remote-as 64512` – indica o IP e o ASN do vizinho. Se indicar o ASN do próprio AS, é porque trata-se de uma sessão iBGP;
- `neighbor 2001:DB8:200:FFFF::255 description R03` – apresenta um nome de identificação;

– `neighbor 2001:DB8:200:FFFF::255 ebgp-multihop 2` – especifica o número de saltos até se alcançar o vizinho. Uma diferença importante entre o iBGP e o eBGP, é que o quando o roteador gera uma mensagem eBGP, o pacote IP que carrega essa mensagem é enviado com o valor do TTL, se for IPv4, ou do *Hop_Limit*, se for IPv6, igual a 1, e com isso, ele só poderá alcançar um salto. Se um pacote tiver que ser roteado para uma interface *loopback*, e a mensagem sair como o TTL igual a 1, o roteador de destino ao abri-lo, irá decrementar o valor do TTL e não poderá realizar o roteamento interno para *loopback*. Por tanto, para levantar a sessão eBGP entre *loopbacks*, deve-se especificar qual o número de saltos;

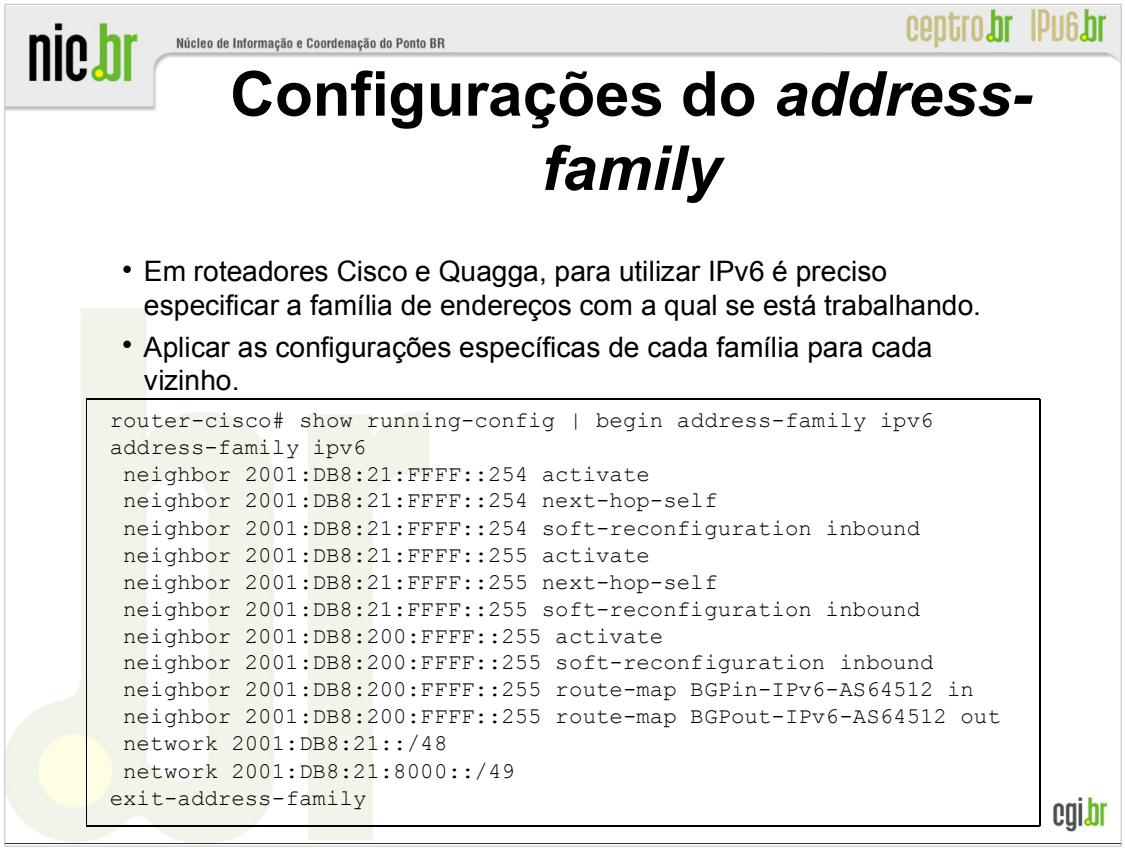
– `neighbor 2001:DB8:200:FFFF::255 update-source Loopback30` - configura o roteador para que o IP de origem dos pacotes seja o da *loopback,* porque ao enviar uma mensagem, o roteador adota por padrão, como endereço IP de origem o IP da interface por onde ela é enviada.

– `neighbor 2001:DB8:200:FFFF::255 version 4` – indica a versão de protocolo BGP utilizada. Essa informação agiliza o estabelecimento da sessão BGP, visto que, na primeira mensagem trocada entre os vizinhos, são passadas algumas informações, entre elas, há a negociação da versão. Se já for informado desde o início qual a versão utilizada, essa negociação não necessita ser feita.

Em relação a configuração do iBGP, as únicas diferenças são que o ASN é o do próprio AS e que diferente do eBGP, não precisa alterar o TTL. Por padrão, no iBGP o TTL não é alterado, presumindo que a sessão possa fazer um caminho longo, saindo com TTL igual a 255 ou outro valor intermediário dependendo da implementação.

Observe a seguir um exemplo das configurações de uma sessão BGP IPv4:

```
router-R13#show running-config | begin bgp
router bgp 64501
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.2.255.255 remote-as 64512
 neighbor 10.2.255.255 description R03
 neighbor 10.2.255.255 ebgp-multihop 2
 neighbor 10.2.255.255 update-source Loopback30
 neighbor 10.2.255.255 version 4
 neighbor 172.21.15.254 remote-as 64501
 neighbor 172.21.15.254 description R12
 neighbor 172.21.15.254 update-source Loopback20
 neighbor 172.21.15.254 version 4
 neighbor 172.21.15.255 remote-as 64501
 neighbor 172.21.15.255 description R11
 neighbor 172.21.15.255 update-source Loopback20
 neighbor 172.21.15.255 version 4
 ...
```



Em roteadores Cisco e Quagga, para utilizar IPv6 é preciso especificar a família de endereços com a qual se está trabalhando. Diferente dos roteadores Juniper, as configurações do BGP são apresentadas dividas em configurações geais e nas configurações específicas de cada família para cada vizinho.

Para analisar as configurações do *address-family* IPv6 utiliza-se:

```
show running-config | begin address-family ipv6
```

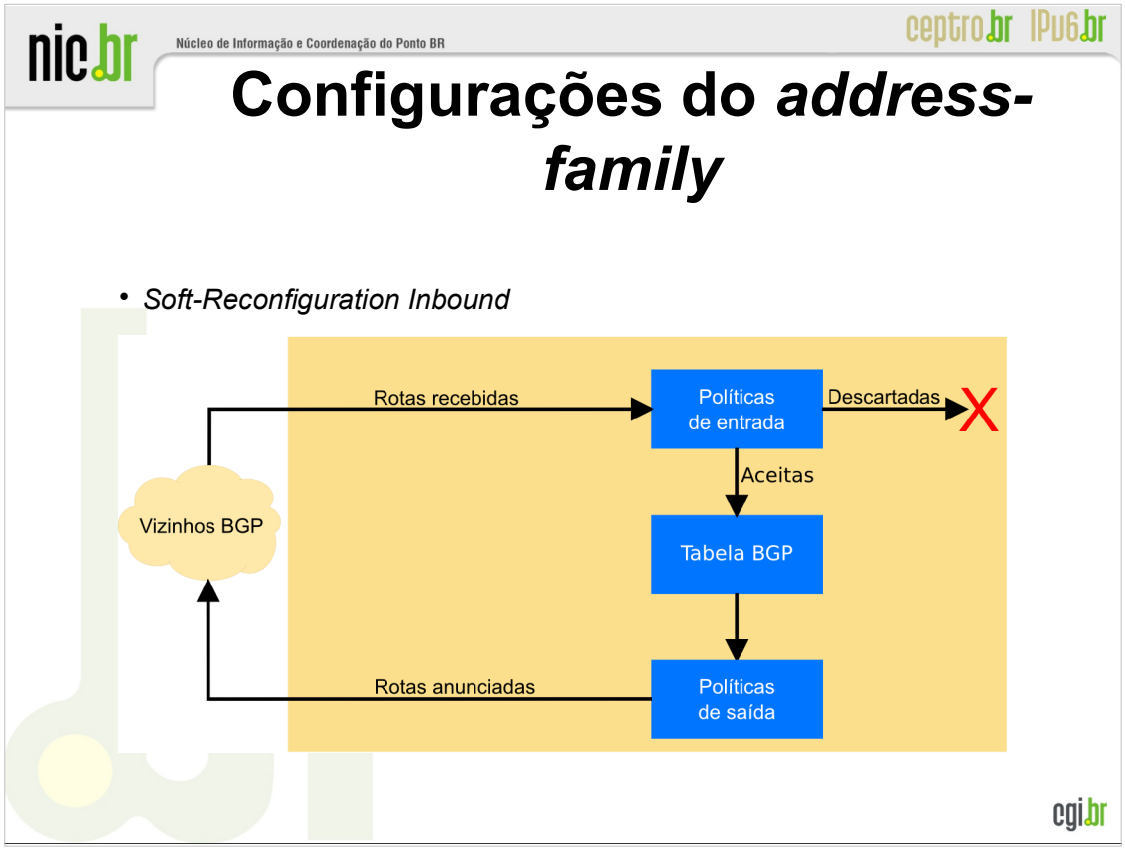
- `address-family ipv6` – indica a qual família pertence as configurações;
- `neighbor 2001:DB8:200:FFFF::255 activate` – ativa a sessão, necessário quando se trabalha com *address-family*. Uma pratica boa a se aplicar quando se configura uma sessão iBGP ou eBGP, é levantá-la em *shutdown*. Isso evita que se estabeleça a sessão sem que as políticas estejam configuradas, não permitindo que se envie informações indevidas;
- **`neighbor 2001:DB8:200:FFFF::255 soft-reconfiguration inbound`** – indica a forma que a tabela de rotas será atualizada;
- **`neighbor 2001:DB8:200:FFFF::255 prefix-list BGPout-IPv6-AS64512 out`** – indica a política de saída aplicada;
- **`neighbor 2001:DB8:200:FFFF::255 route-map BGPin-IPv6-AS64512 in`** – indica a política de saída aplicada.

Em relação as configurações do iBGP destaca-se a seguinte informação:

- `neighbor 2001:DB8:21:FFFF::254 next-hop-self` – indica quem é o próximo salto. Esta configuração ajuda a trazer mais estabilidade e facilita na operação dos ASs. Um roteador de borda pode repassar para os demais roteadores de seu AS, via iBGP, todos os prefixos que ele aprender de seus ASs vizinhos. Quando ele repassa para os demais roteadores, é mantido o atributo *next-hop*. No entanto, o *next-hop* desses prefixos sempre será o roteador de borda dos ASs vizinhos, aos quais os roteadores internos não possuem conectividade direta. Para solucionar esse problema, o roteador de borda do AS repassa os anúncios informando que o *next-hop* para os ASs vizinhos é ele mesmo através do comando **`next-hop-self`**. Com isso, os roteadores internos só precisam saber chegar no roteador de borda de seu AS, que é quem tem conectividade para a Internet.

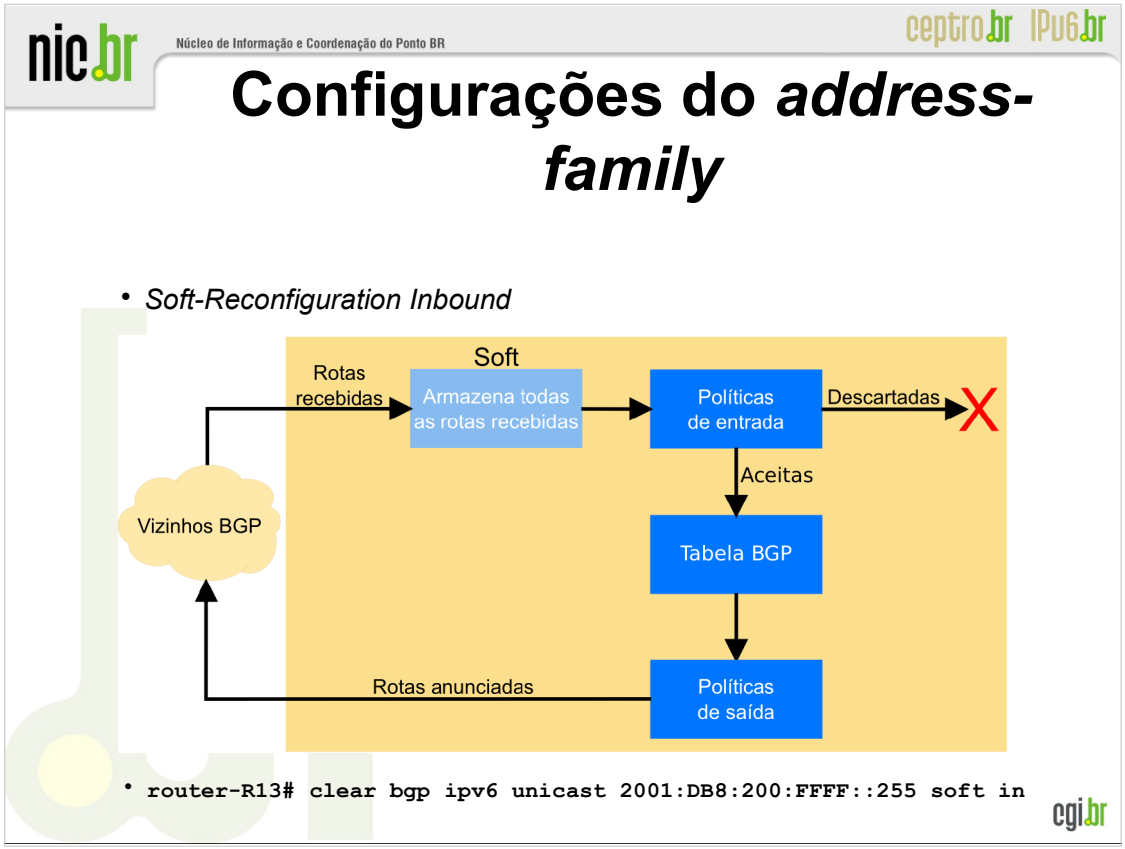
Um exemplo de configuração do *addres-family* IPv4 pode ser observado a seguir:

```
router-cisco# show running-config | begin address-family ipv4
address-family ipv4
 neighbor 10.2.255.255 activate
 neighbor 10.2.255.255 soft-reconfiguration inbound
 neighbor 10.2.255.255 prefix-list BGPout-IPv4-AS64512 out
 neighbor 10.2.255.255 route-map BGPin-IPv4-AS64512 in
 neighbor 172.21.15.254 activate
 neighbor 172.21.15.254 next-hop-self
 neighbor 172.21.15.254 soft-reconfiguration inbound
 neighbor 172.21.15.255 activate
 neighbor 172.21.15.255 next-hop-self
 neighbor 172.21.15.255 soft-reconfiguration inbound
 network 172.21.0.0 mask 255.255.240.0
 network 172.21.8.0 mask 255.255.248.0
exit-address-family
```



Um comando interessante é o `soft-reconfiguration inbound`. Analise o seguinte exemplo:

- O roteador R1 levanta uma sessão BGP com o roteador R2;
- Quando a sessão é estabelecida, o roteador envia todas as informações que ele conhece;
- Novas informações só serão enviadas quando houver a necessidade de se adicionar ou retirar entradas da tabela;
- Se for criada uma política de entrada no R2, a mensagem original, trocada no estabelecimento da sessão, será alterada;
- Caso seja necessário criar uma nova política em R2, não se terá mais as informações iniciais para poder aplicá-las;



Uma forma de se recuperar essas informações seria "derrubar" a sessão, para que assim o roteador mande novamente todos os seus prefixos. Essa prática era funcional quando não havia tantas entradas na tabela global de rotas. Hoje em dia ela não é mais efetiva.

Outra opção é utilizar o comando `soft-reconfiguration inbound`. Com isso, antes de se aplicar as políticas é criada uma outra tabela de entrada por vizinho, exatamente igual a recebida. Deste modo, tudo o que o R1 enviar, será gravado nesta pré-tabela, salvando os prefixos originais. Se for necessário alterar alguma configuração, basta utilizar, por exemplo, o comando:

```
router-R13# clear bgp ipv6 unicast 2001:DB8:200:FFFF::255 soft in
```

Este comando faz com que o roteador releia a pré-tabela sem interromper a sessão. Porém, isso também não é funcional nos dias de hoje, porque a tabela BGP possui entorno de 300 mil prefixos, e com esse comando duplica-se a tabela BGP para cada vizinho, consumindo muito mais memória do seu módulo de roteamento, na parte de controle.

Um exemplo de sua utilização com IPv4 seria:

```
router-R13# clear bgp ipv4 unicast 10.2.255.255 soft in
```

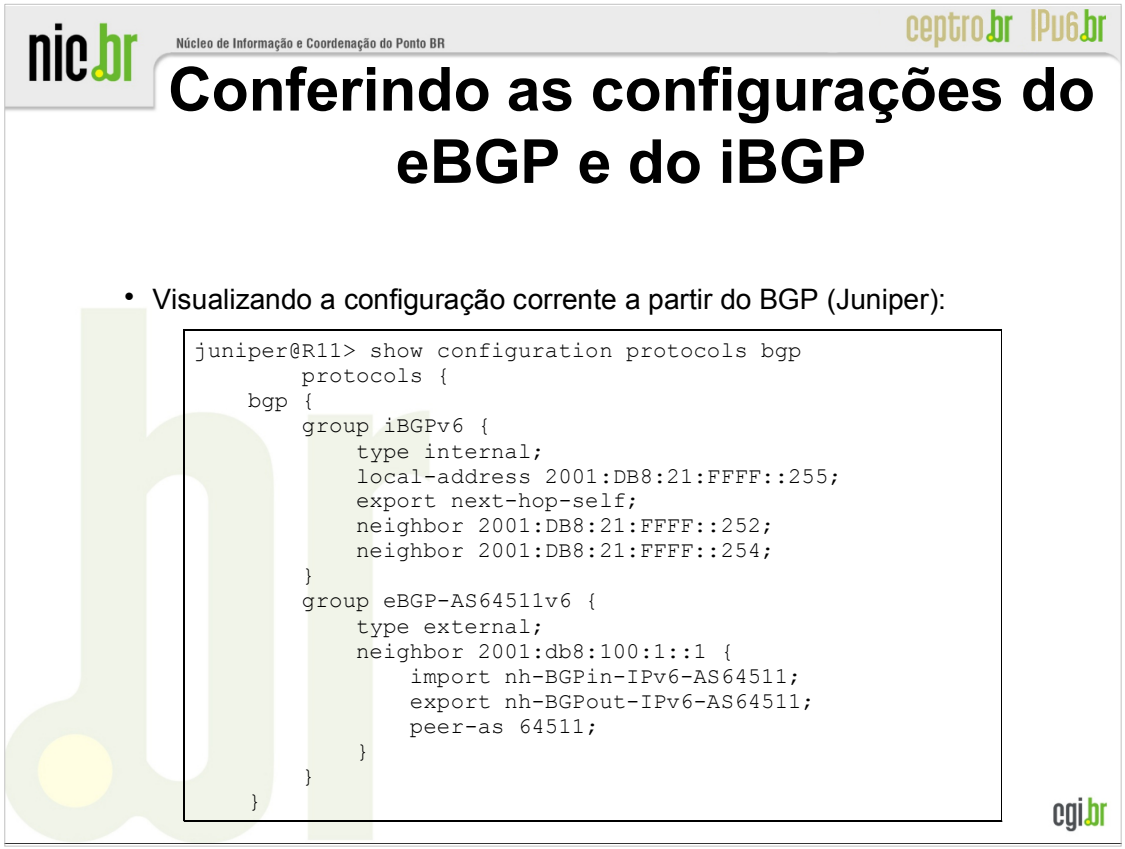
Quando os roteadores iniciam uma sessão BGP, cada roteador passa uma série de informações sobre os recursos que ele conhece, como: quais *capabilities* ele suporta. Uma delas é o `route-refresh`.

Este recurso permite recuperar as informações originais da tabela de rotas sem "derrubar" a sessão BGP e sem criar tabelas adicionais, apenas solicitando ao vizinho o reenvio da tabela de rotas.

Para saber se o roteador suporta `route-refresh` um exemplo seria:

```
show ipv6 bgp neighbor 2001:DB8:200:FFFF::255
show ip bgp neighbor 10.2.255.255
```

Com este comando também é possível ver o suporta a outras *capabilities* como o suporte a ASN de 32 bits (*New ASN Capability*).



Roteadores Juniper já trabalham com o conceito de *address-family* por padrão. (inet, inet6).

Para visualiza a configuração corrente a partir do BGP em um roteador Juniper:

```
show configuration protocols bgp
```

No primeiro grupo são apresentadas as configurações do iBGP informando os seguintes dados:

- `group iBGPv6` – nome do grupo;
- `type internal` – indica que é iBGP;
- `local-address 2001:DB8:21:FFFF::255` – endereço da interface de saída;
- `export next-hop-self` – propaga aos roteadores internos que o próximo salto para qualquer rota é o roteador de borda do próprio AS;
- `neighbor 2001:DB8:21:FFFF::252` – indica o IP do vizinho iBGP;
- `neighbor 2001:DB8:21:FFFF::254` - indica o IP do vizinho iBGP.

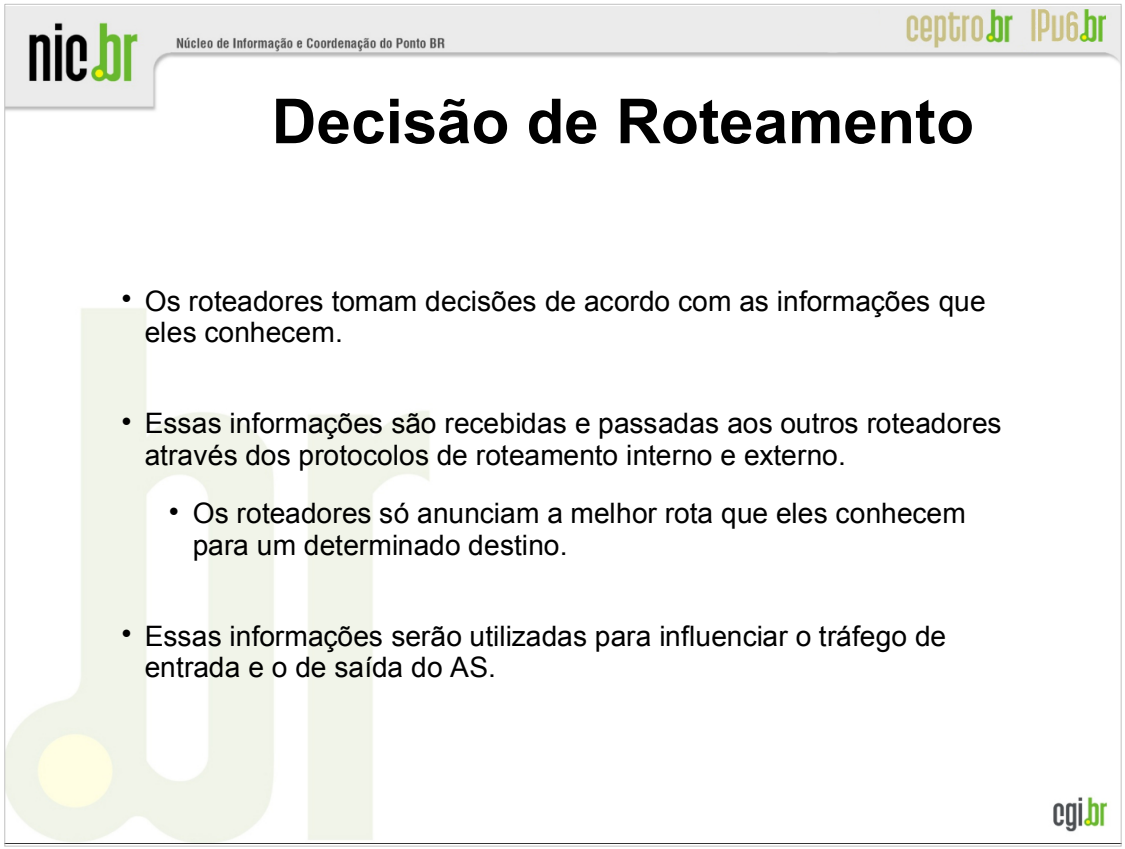
O segundo grupo traz as informações do eBGP:

- `group eBGP-AS64511` – nome do grupo;
- `type external` – indica que é eBGP;
- `neighbor 2001:db8:100:1::1` – indica o endereço do vizinho eBGP;
- `import nh-BGPin-IPv6-AS64511` – política de entrada aplicada;
- `export nh-BGPout-IPv6-AS64511` – política de saída aplicada;
- `peer-as 64511` – ASN do vizinho.

Diferente da configuração do roteador Cisco apresentada anteriormente, no exemplo acima foi utilizado o endereço IP da interface real para se estabelecer as sessões BGP.

Observe a seguir um exemplo das configurações de uma sessão BGP IPv4 no Juniper:

```
juniper@R11> show configuration protocols bgp
protocols {
    bgp {
        group iBGP {
            type internal;
            local-address 172.21.15.255;
            export next-hop-self;
            neighbor 172.21.15.252;
            neighbor 172.21.15.254;
        }
        group eBGP-AS64511 {
            type external;
            neighbor 10.1.1.1 {
                import nh-BGPin-IPv4-AS64511;
                export nh-BGPout-IPv4-AS64511;
                peer-as 64511;
            }
        }
    }
```



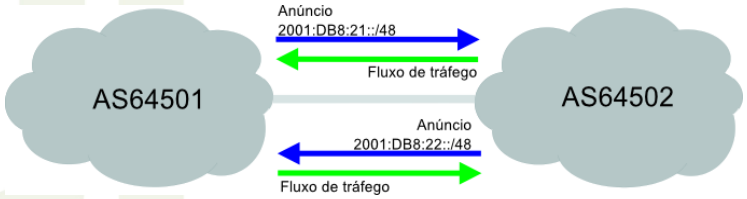
Os roteadores tomam decisões de acordo com as informações que eles conhecem. Essas informações são recebidas e passadas aos outros roteadores através dos protocolos de roteamento interno e externo.

Ao enviar suas informações, os roteadores só anunciam a melhor rota que eles conhecem para um determinado destino. São essas informações que serão utilizadas para influenciar o tráfego de entrada e o de saída do AS.

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

Influenciando o Tráfego

- Os prefixos que um AS anuncia, interferem no tráfego de entrada ou saída?
  - Os prefixos anunciados interferem na forma como os outros conhecem o AS.
    - tráfego de entrada.
  - Os prefixos recebidos de outras redes interferem no tráfego de saída.



cgi.br

Os prefixos anunciados interferem na forma como os outros conhecem o AS, isso é, interferem no tráfego de entrada. Do mesmo modo, os prefixos recebidos interferem no tráfego de saída.

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

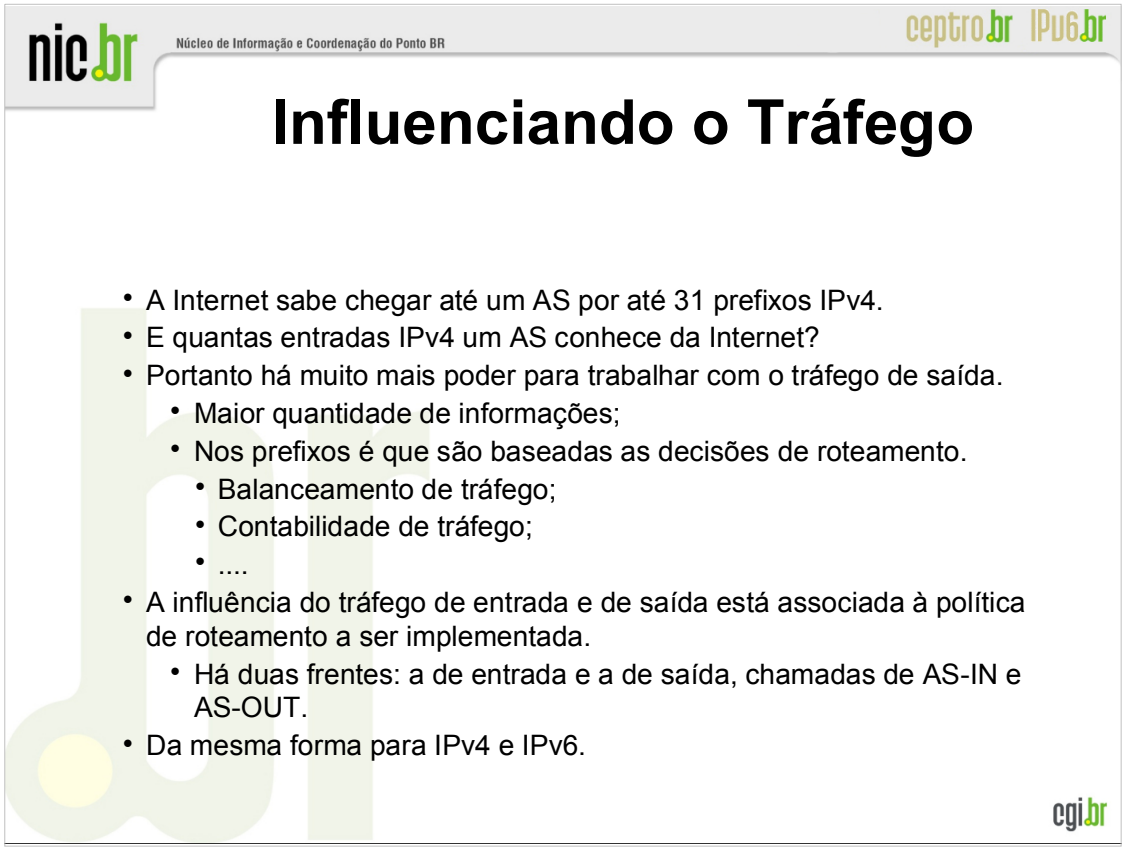
Influenciando o Tráfego

- O que é mais fácil, influenciar o tráfego de entrada ou de saída?
- Ex.:
  - Um AS possui um bloco IPv4 /20;
    - Este AS pode gerar para a Internet anúncios de prefixos até um /24, o prefixo IPv4 mais específico normalmente aceito pelas operadoras;
    - Quantos prefixos /24 podem ser gerados a partir de um /20?
    - E quantos prefixos podem ser gerados entre /20 e um /24?
    - E entre um /32 e um /48 IPv6?

cgi.br

O que é mais fácil em um AS, influenciar o tráfego de entrada ou de saída?
Analise as informações do exemplo a seguir:

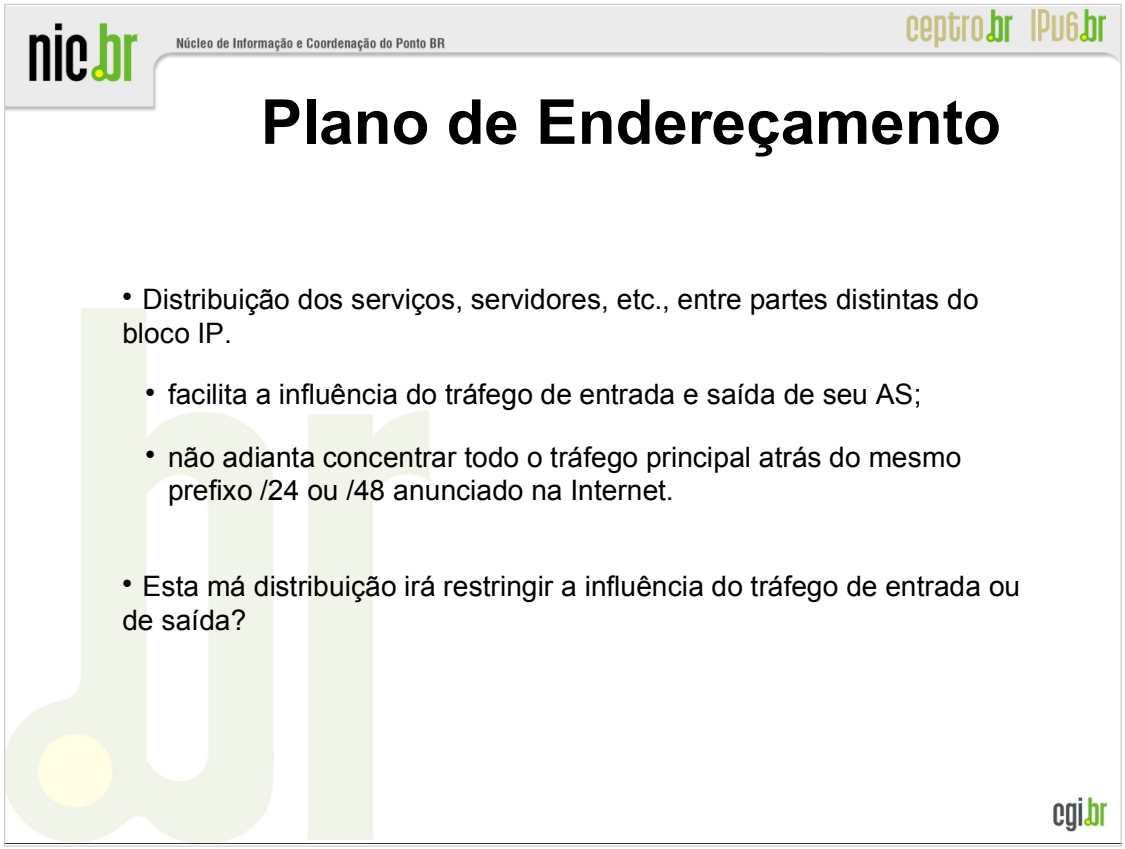
- Um AS possui um bloco IPv4 /20;
  - Este AS pode gerar para a Internet anúncios de prefixos até um /24, o prefixo IPv4 mais específico normalmente aceito pelas operadoras;
  - Com um /20 pode-se gerar 16 prefixos /24;
  - Se considerarmos a hipótese de se anunciar todos os prefixos possíveis entre o /20 até o /24, pode-se gerar um total de 31 prefixos;
  - E com um bloco de endereços IPv6, quantos prefixos podem ser gerados entre um /32 e um /48?



Deste modo, a Internet sabe chegar até o AS por até 31 prefixos IPv4 e o AS tem a opção de sair por aproximadamente 360.000 prefixos, que é o tamanho da Tabela de Roteamento Global IPv4 atualmente.

Portanto, se pensarmos na ideia de que “informação é poder”, podemos afirmar que é mais fácil influenciar o tráfego de saída, visto que há uma maior quantidade de prefixos para se trabalhar, e que são nos prefixos que são baseadas as decisões de roteamento, atuando sobre o balanceamento, contabilidade de tráfego etc.

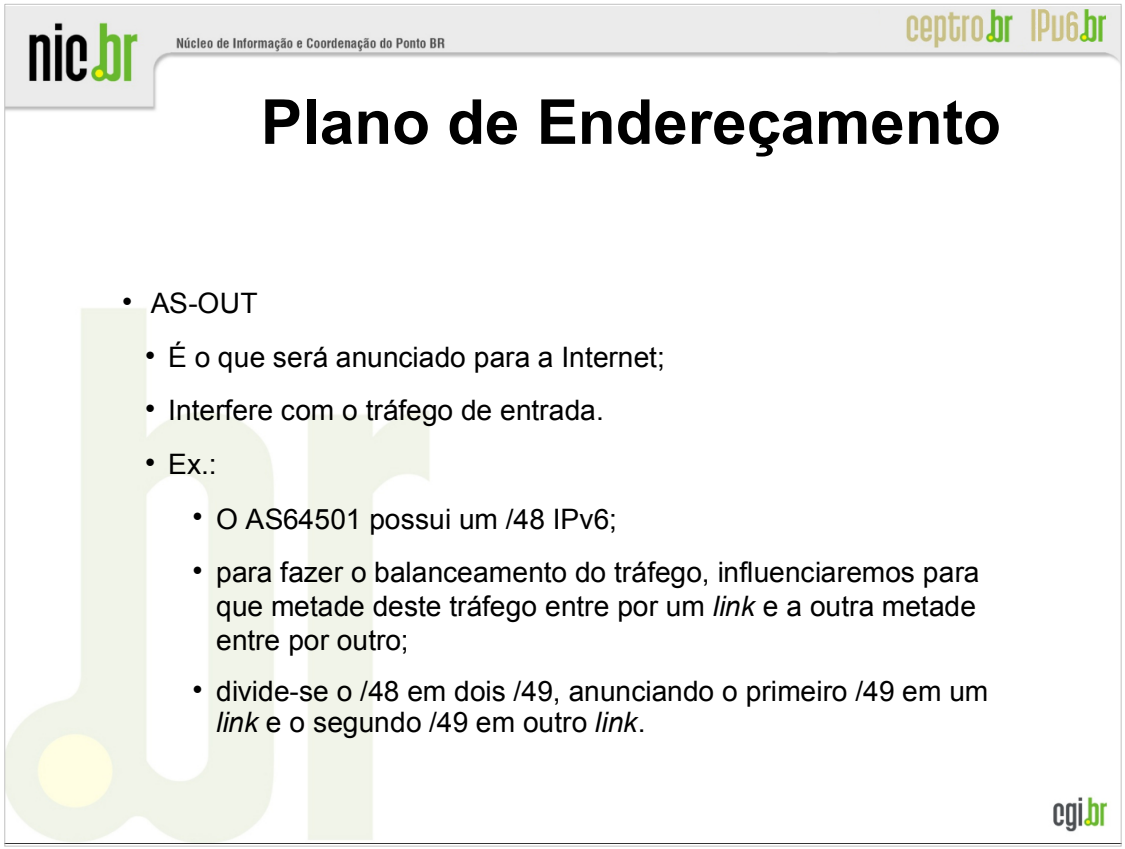
A influência dos tráfegos de entrada e de saída está associada à política de roteamento a ser implementada, sendo que há duas frentes: a de entrada e a de saída, chamadas de AS-IN e AS-OUT. Isto ocorre da mesma forma para IPv4 e IPv6.



Um ponto importante no plano de endereçamento IP, tanto IPv4 quanto IPv6, é a distribuição dos serviços, servidores, etc., entre partes distintas do bloco IP, de modo a facilitar a influência do tráfego de entrada e saída do AS.

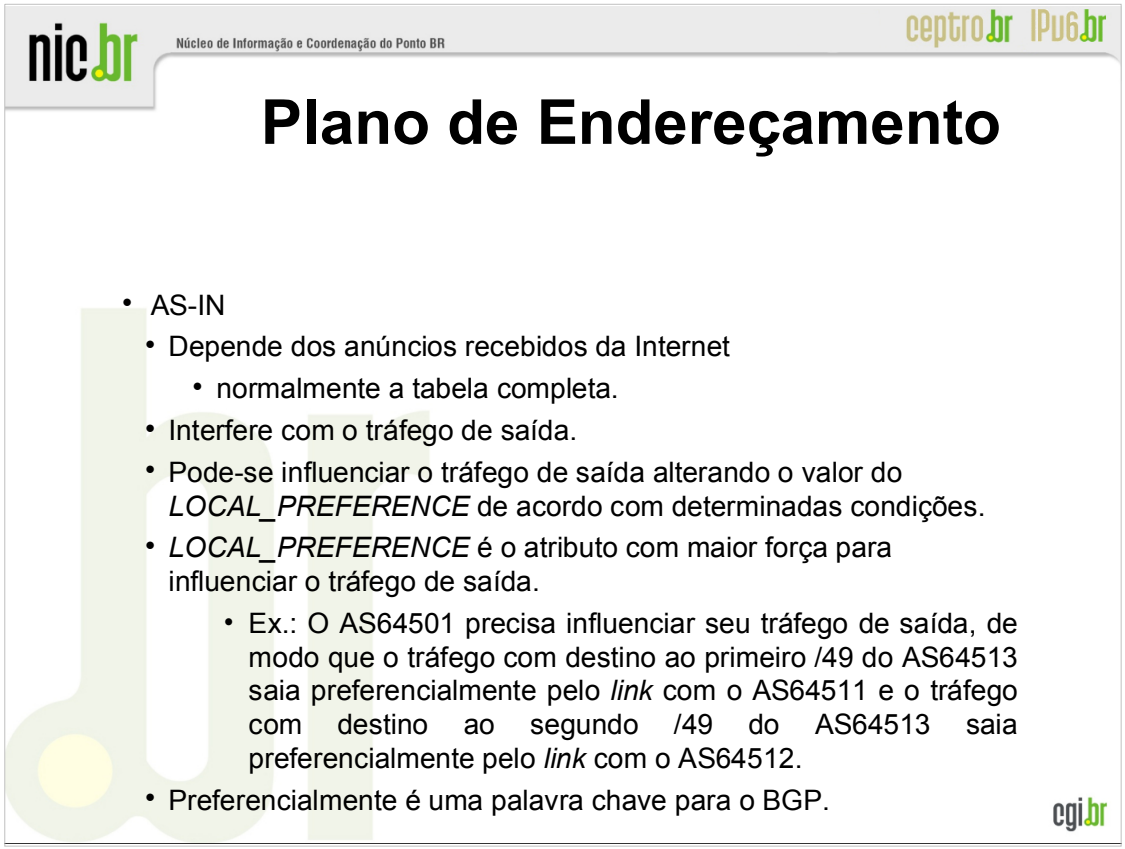
Não é recomendável posicionar todos os servidores, clientes importantes que geram a maior parte do tráfego, no mesmo /24 IPv4 ou /48 IPv6. Esta má distribuição restringirá a influencia do tráfego de entrada. Porque o tráfego de entrada é influenciado pelos anúncios gerados. Ou seja, é fundamental no plano de endereçamento ver como serão posicionados os consumidores de tráfego de entrada e de saída.

Consumidor de tráfego de entrada, que são os usuários de acesso a Internet, tem que se distribuir bem entre os prefixos anunciados.



AS-OUT é a política que trata do que será anunciado para a Internet. É o que vai interferir com o tráfego de entrada.

Por exemplo, o AS64501 possui um bloco /48 IPv6 e, para dividir e fazer o balanceamento do tráfego de entrada por dois links de acesso à Internet, deve-se influenciá-lo para que uma metade entre por um *link* e a outra metade entre por outro *link.* Para isso, divide-se o /48 em dois prefixos /49, anunciando o primeiro /49 em um *link* e o segundo /49 em outro *link*. Isto é utilizado para possibilitar um balanceamento do tráfego.

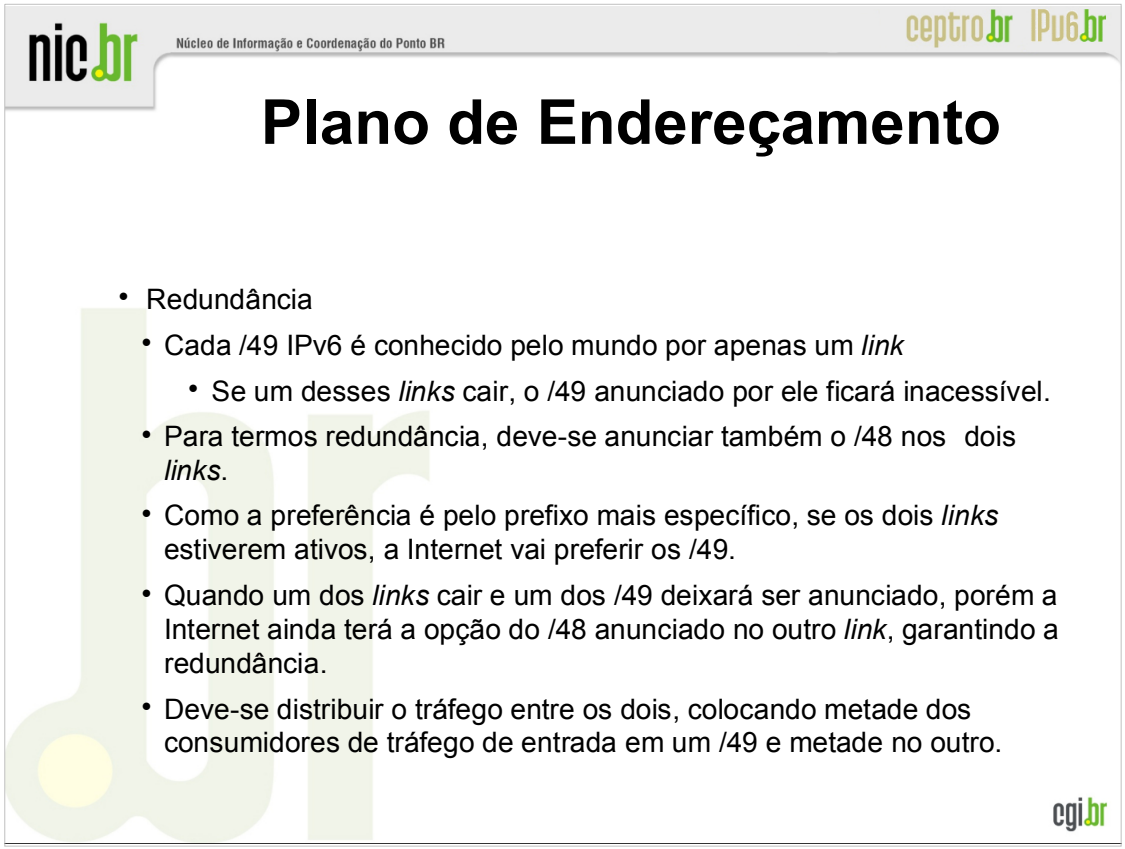


AS-IN é a política que irá interferir no tráfego de saída. Ela depende dos anúncios que são recebidos da Internet, normalmente a tabela de roteamento completa.

Para influenciar o tráfego de saída, pode-se alterar o valor do atributo *LOCAL_PREFERENCE* dos anúncios recebidos, de acordo com determinadas condições. Ele é o atributo com maior força para influenciar este tipo de tráfego.

> Ex.: O AS 64501 precisa influenciar seu tráfego de saída, de modo que o tráfego com destino ao primeiro /49 do AS64513 saia preferencialmente pelo *link* com o AS64511 e o tráfego com destino ao segundo /49 do AS64513 saia preferencialmente pelo *link* com o AS64512.

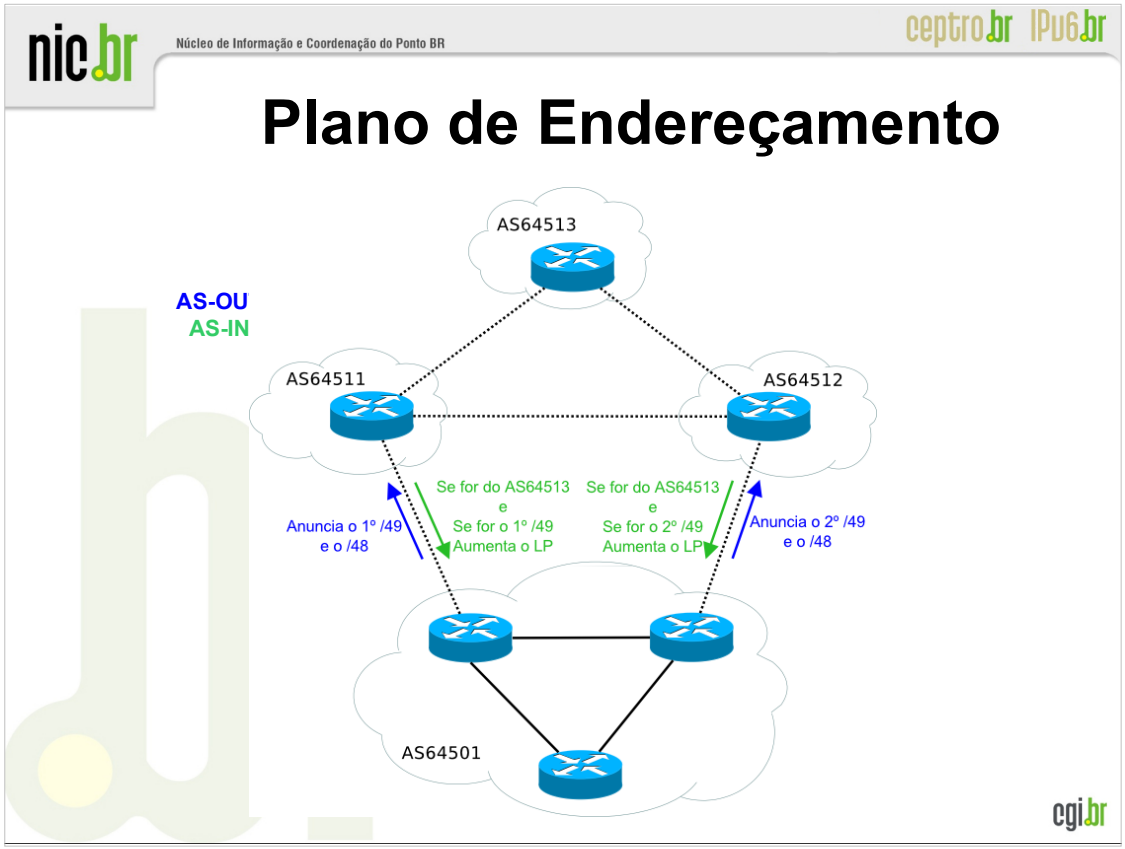
Preferencialmente é uma palavra chave para o BGP. É fundamental, com BGP, trabalhar com preferência e não com filtro. Existem várias formas de se fazer isso, mas, para que seja realmente efetiva uma configuração, especialmente quando ocorre uma queda de *link*, é importante não se descartar nada, deve-se preferenciar um caminho em relação ao outro, mas não descartar. O descarte tem um efeito imediato, mas quando não houver redundância, ele deixa de ser efetivo.



Para que haja redundância deve-se atuar da seguinte forma:

- Após dividir o bloco /48 IPv6 em dois prefixos /49, cada um deles é anunciado em um *link*, isso é, cada /49 é conhecido pelo mundo por apenas um *link*, um caminho;
- Se um desses *links* cair, o /49 anunciado por ele ficará inacessível;
- Para termos redundância, deve-se anunciar também o prefixo /48 nos dois *links;*
- Com isso, a Internet conhecerá o /48 pelos dois caminhos;
- Como a preferência é pelo prefixo mais específico, se os dois *links* estiverem ativos, a Internet vai preferir os /49 sempre, ou seja o balanceamento estará em operação;
- Quando um desses *links* cair, um dos prefixos /49 deixará de ser anunciado. No entanto, esse /49 está contido no /48, ou seja, mesmo com um dos *links* desativado, a Internet ainda terá a opção do /48 anunciado no outro *link*. Garantindo a redundância.

Para realmente distribuir o tráfego entre os dois, deve-se colocar metade dos consumidores de tráfego de entrada em um /49 e metade no outro. Isso faz parte do planejamento.

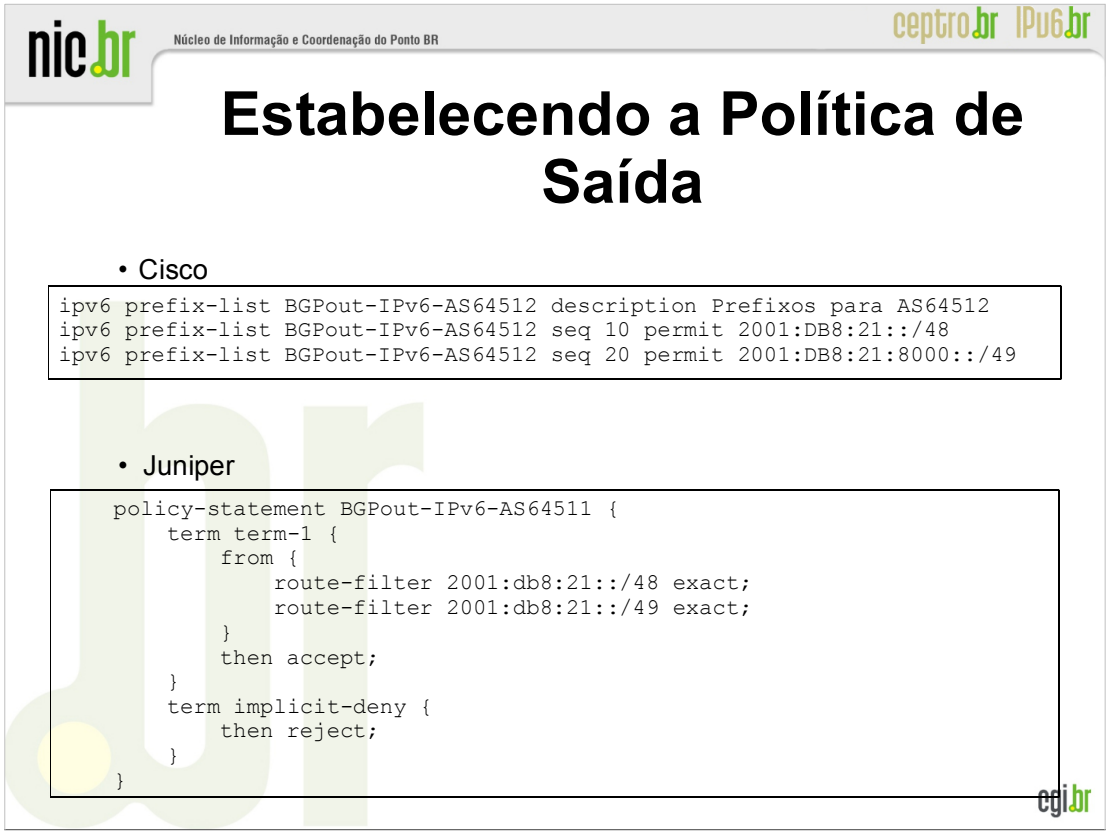


Neste diagrama podemos analisar exemplos das políticas de roteamento apresentadas até o momento.

- AS-OUT – Utiliza os anúncios enviados para interferir o tráfego de entrada;
- AS-IN – Utiliza os anúncios recebidos para influenciar o tráfego de saída.

Uma função importante do BGP reside na manipulação dos atributos e nos testes condicionais. Para tratar esses aspectos temos as seguintes funcionalidades:

- *route-map* (Cisco e Quagga) – define as condições para a redistribuição de rotas e permite controlar e modificar informações de políticas de roteamento;
- *prefix-list* (Cisco e Quagga) – mecanismo de filtragem de prefixos com muitos recursos. Permite trabalhar com notação de prefixo, adicionar descrição e trabalhar com sequencia, apresentando deste modo, uma vantagem em relação a utilização da função *distribute-list,* que por ser baseada em ACLs facilita a filtragem de pacotes, porém, torna-se de difícil gerenciamento;
- *route-filter* (Juniper) – utilizado para comparar rotas individualmente ou em grupos.



Este exemplo mostra através de *prefix-list* em um roteador Cisco e de *route-filter* em um roteador Juniper, o estabelecimento de uma política de saída.

No exemplo do roteador Cisco, os *prefix-list* indicam os prefixos que serão anunciados para o AS64512. Neste caso serão enviados o prefixo /48 IPv6 e segundo /49.

No exemplo do roteador Juniper, os *route-filter* indicam que serão anunciados para o AS 64511 o prefixo /48 IPv6 e o primeiro /49.

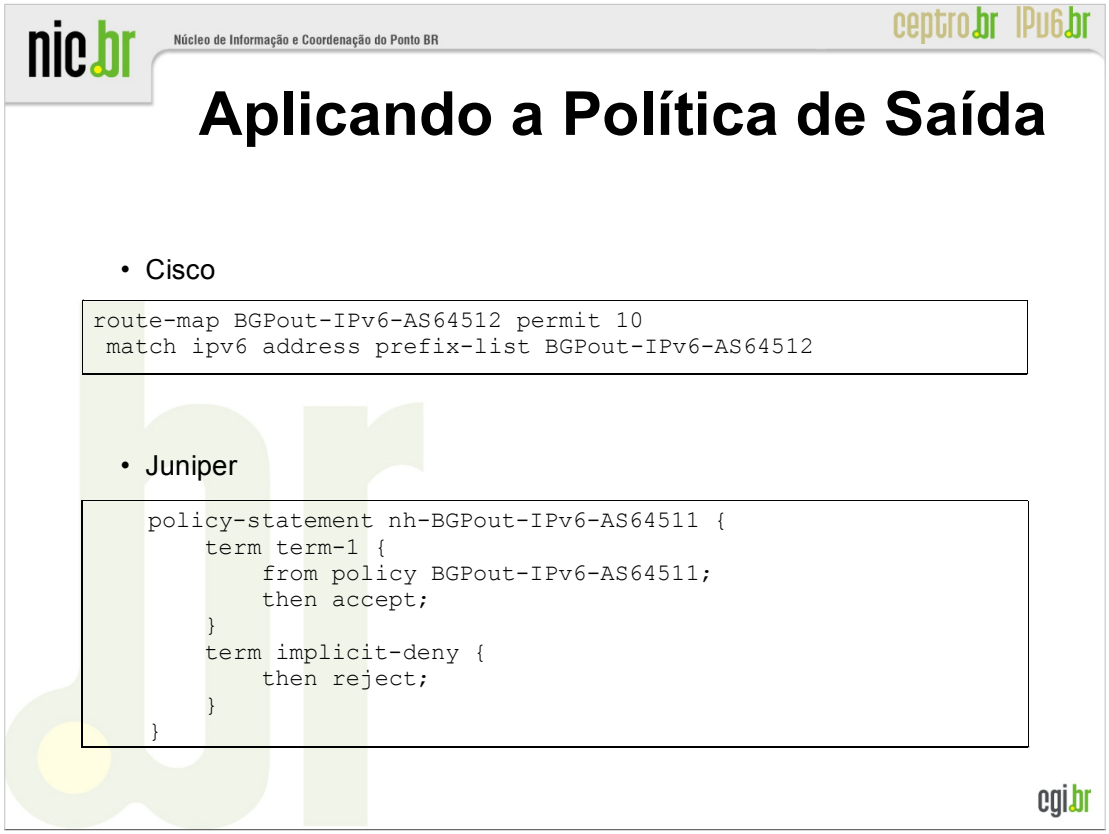
Um exemplo dessa política de saída em uma configuração IPv4 poderia ser:

- Cisco

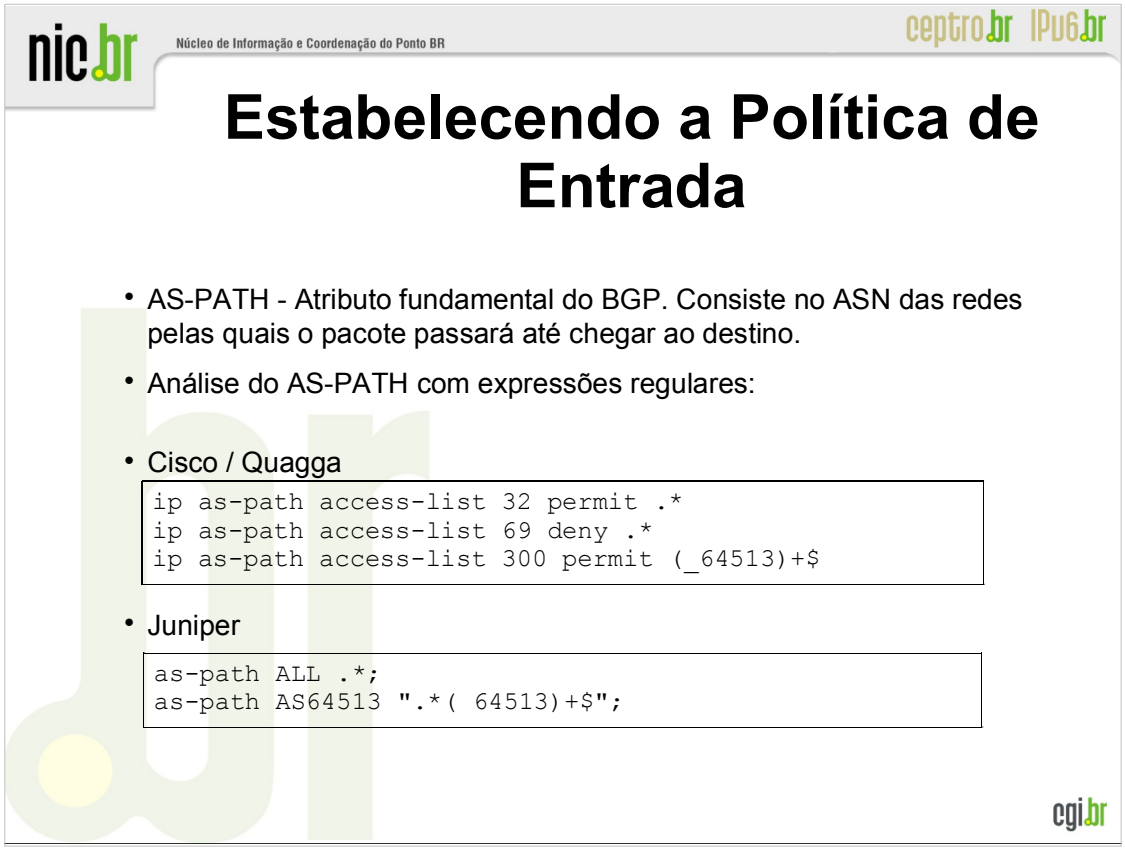
```
ip prefix-list BGPout-IPv4-AS64512 seq 10 permit 172.21.0.0/20
ip prefix-list BGPout-IPv4-AS64512 seq 20 permit 172.21.8.0/21
```

- Juniper

```
route-filter 172.21.0.0/20 exact;
route-filter 172.21.0.0/21 exact;
```



Do mesmo modo como foram aplicadas as políticas de entrada, podemos aplicar as políticas de saída através de um `route-map` e de um `policy-statement` no Juniper.



O BGP é um protocolo usado na comunicação entre os ASs. Por isso, o AS-PATH torna-se um atributo fundamental do BGP. Ele consiste no ASN das redes pelas quais o pacote passará até chegar ao destino.

Os roteadores Cisco, Quaga e Juniper oferecem comandos que permitem a análise do AS-PATH com expressões regulares. Em nosso exemplo, temos os seguintes termos nas expressões regulares:

- O caractere “ponto” significa 'qualquer elemento'
- O caractere “asterisco” significa 'zero ou varias ocorrências'
- O caractere “$” significa 'fim de linha'
- O caractere “+” significa 'uma ou mais ocorrências'

`ip as-path access-list 32 permit .*` (Cisco / Quagga)
`as-path ALL .*;` (Juniper)

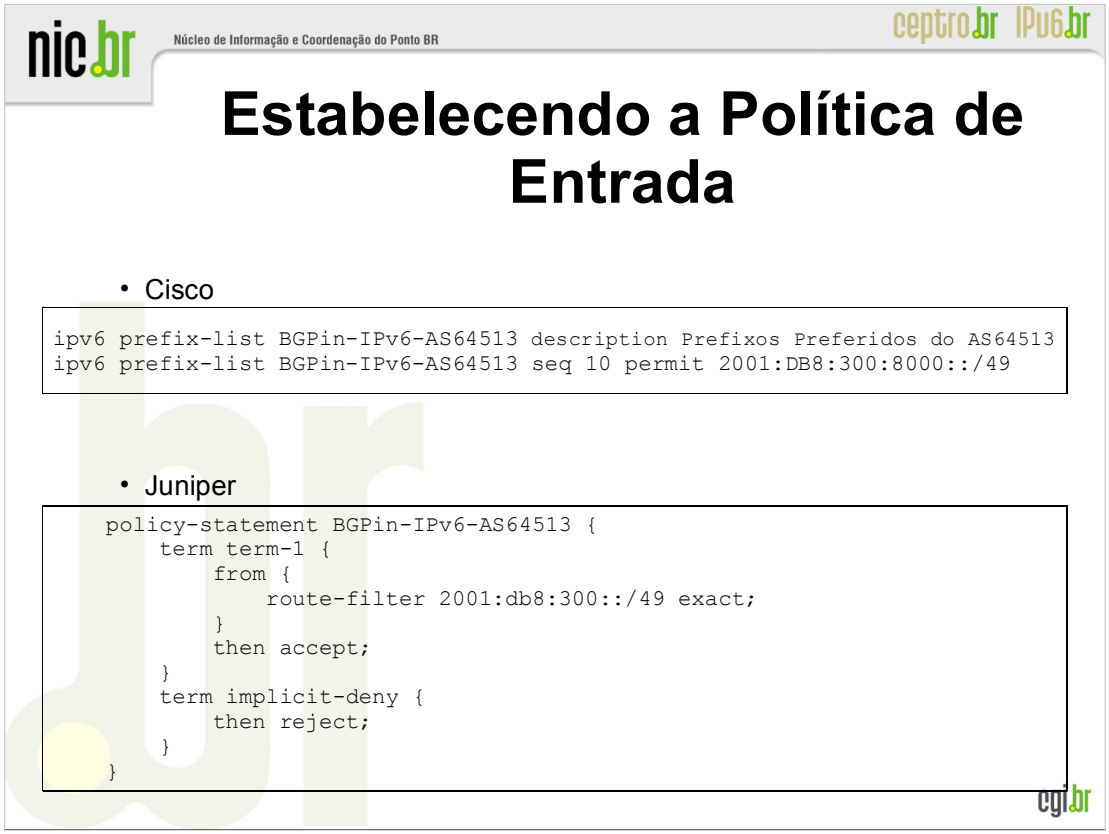
- As linhas acima indicam que qualquer AS-PATH será permitido.

`ip as-path access-list 69 deny .*` (Cisco / Quagga)

- Esta linha indica que qualquer bloco será negado.

`ip as-path access-list 300 permit (_64513)+$` (Cisco / Quagga)
`as-path AS64513 ".*( 64513)+$";` (Juniper)

- Essas linhas indicam que todos os prefixos originados no AS 64513 serão permitidos. A permissão de uma ou mais ocorrências é para garantir *AS Path prepends*, ou seja, repetições de um mesmo ASN em sequencia ao longo do AS-PATH (no caso, o ASN 64513).



Este exemplo mostra através de um *prefix-list* em um roteador Cisco e um *route-filter* em um roteador Juniper, o estabelecimento de uma política de entrada.

No exemplo do roteador Cisco, o *prefix-list* identifica o segunda /49 do AS 64513 que será recebido.

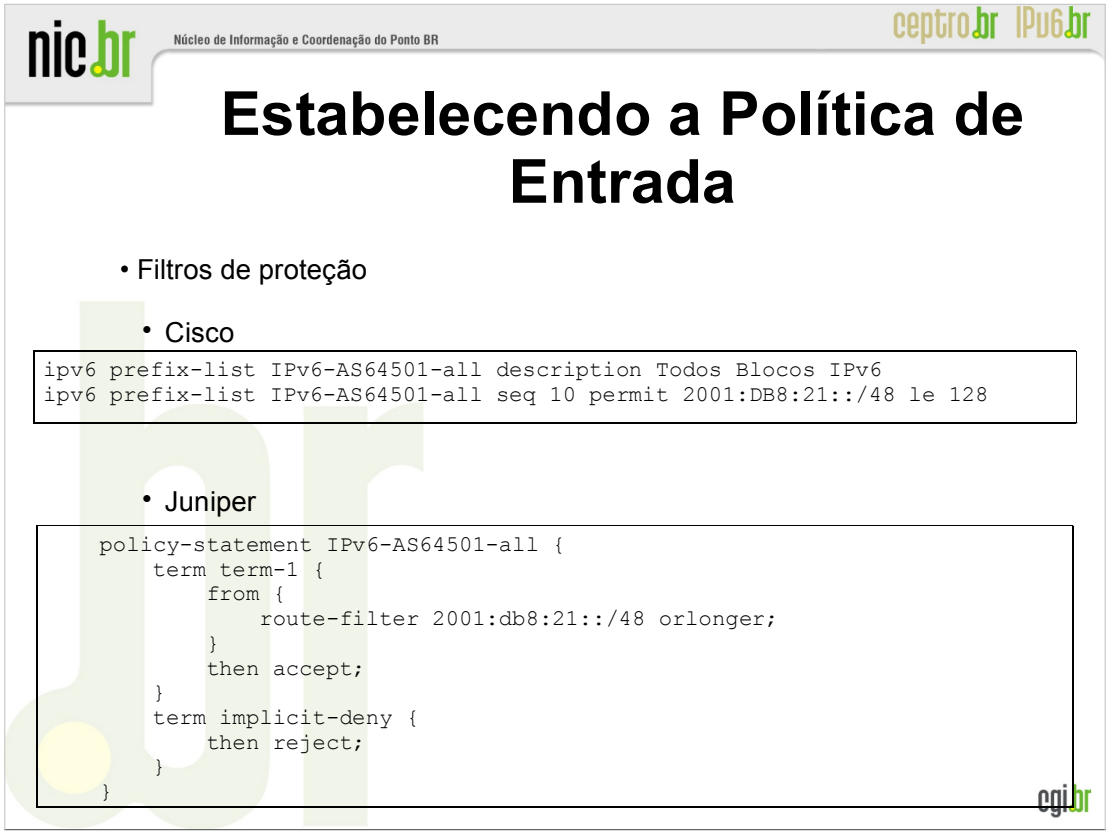
```
ipv6 prefix-list BGPin-IPv6-AS64513 seq 10 permit 2001:DB8:300:8000::/49
```

No exemplo do roteador Juniper, o *route-filter* identifica o primeiro /49 do AS 64513 que será recebido.

```
route-filter 2001:db8:300::/49 exact;
```

Um exemplo da implantação dessas políticas em uma configuração IPv4 seria:

- Cisco: `ip prefix-list BGPin-IPv6-AS64513 seq 10 permit 10.3.128.0/17`
- Juniper: `route-filter 10.3.0.0/17 exact;`



Outra política de proteção importante é a que impede que o AS receba anúncios de seus próprios prefixos.

Para IPv6 poderíamos ter algo similar a:

- Cisco/Quagga

```
ipv6 prefix-list IPv6-AS64501-all seq 10 permit 2001:DB8:21::/48 le 128
```

- Juniper

```
route-filter 2001:db8:21::/48 orlonger;
```

Para IPv4 poderíamos ter algo similar a:

- Cisco / Quagga

```
ip prefix-list IPv4-AS64501-all seq 10 permit 172.21.0.0/20 le 32
```

- Juniper

```
route-filter 172.21.0.0/20 orlonger;
```

As regras exemplificadas acima indicam todos os prefixos possíveis dentro de um bloco /48 IPv6 ou /20 IPv4. Elas serão utilizadas na política de entrada, para dizer: “não aceito nenhum prefixo que seja meu”. Isso ajuda a evitar problemas como sequestro de prefixos.

Por padrão, o roteador rejeita todos os prefixos que tenha o seu ASN, para evitar *looping*. Porém, nada impede que outro AS na Internet, por intensão ou erro, gere anúncios do outro prefixo, até um mais específico. Se não houver proteção, o roteador vai aceitar e encaminhar todo o tráfego interno para fora.

Um exemplo deste tipo de ocorrência foio sequestro do prefixo do YouTube. Por determinação do Governo Paquistanês, o tráfego do YouTube deveria ser bloqueado para evitar o acesso ao trailer de um filme anti-Islamico. Para cumprir essa ordem, a operadora Pakistan Telecom gerou o anúncio de um prefixo mais específico do que o utilizado pelo YouTube, com o intuito de direcionar todos os acessos a ele para uma página que dizia "YouTube was blocked".

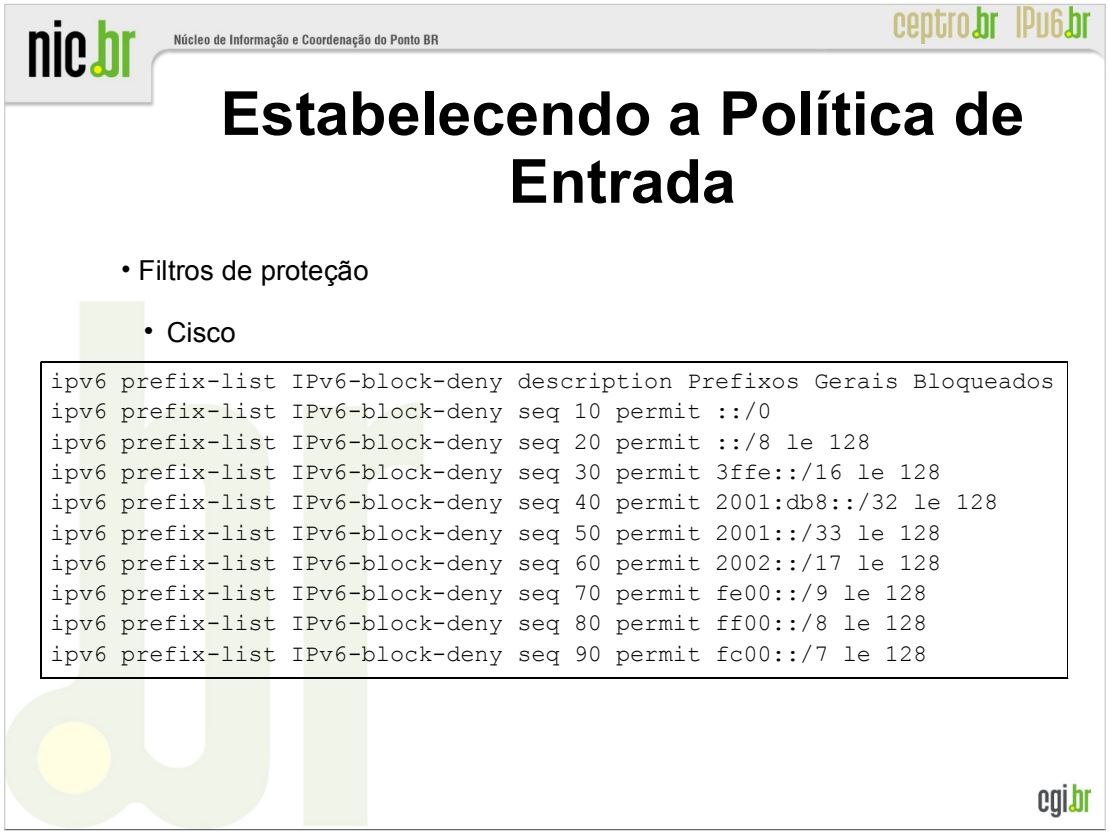
No entanto, a operadora anunciou essa nova rota a seu *upstream provider* (primeiro erro), que, além de não verificar a nova rota (segundo erro) a propagou por toda a Internet (terceiro erro). Com isso, todo o tráfego do YouTube passou a ser direcionado para o Paquistão e ser descartado.

Esse foi apenas o caso mais famoso, mas sequestros de blocos IP ocorrem diariamente, intencionalmente ou não. Isso ocorre porque toda a estrutura da Internet e o funcionamento do BGP foram definidos baseados em uma relação de confiança. Essa "inocência" ainda é presente e é o que mantem toda a estrutura atual.

Existem discussões sobre modos de verificar se o AS que está anunciando um determinado prefixo tem autoridade para faze-lo, similar ao ao que o DNSSec faz.

Mais informações:

- http://www.ietf.org/dyn/wg/charter/idr-charter.html
- http://www.ietf.org/dyn/wg/charter/sidr-charter.html
- http://www.youtube.com/watch?v=IzLPKuAOe50
- http://www.wired.com/threatlevel/2008/02/pakistans-accid/



Também é possível adicionar *prefix-list* de proteção. Observe o seguinte exemplo em um roteador Cisco:

```
ipv6 prefix-list IPv6-block-deny seq 10 permit ::/0
ipv6 prefix-list IPv6-block-deny seq 20 permit ::/8 le 128
ipv6 prefix-list IPv6-block-deny seq 30 permit 3ffe::/16 le 128
ipv6 prefix-list IPv6-block-deny seq 40 permit 2001:db8::/32 le 128
ipv6 prefix-list IPv6-block-deny seq 50 permit 2001::/33 le 128
ipv6 prefix-list IPv6-block-deny seq 60 permit 2002::/17 le 128
ipv6 prefix-list IPv6-block-deny seq 70 permit fe00::/9 le 128
ipv6 prefix-list IPv6-block-deny seq 80 permit ff00::/8 le 128
ipv6 prefix-list IPv6-block-deny seq 90 permit fc00::/7 le 128
```

Estes *prefix-list* verificam respectivamente:

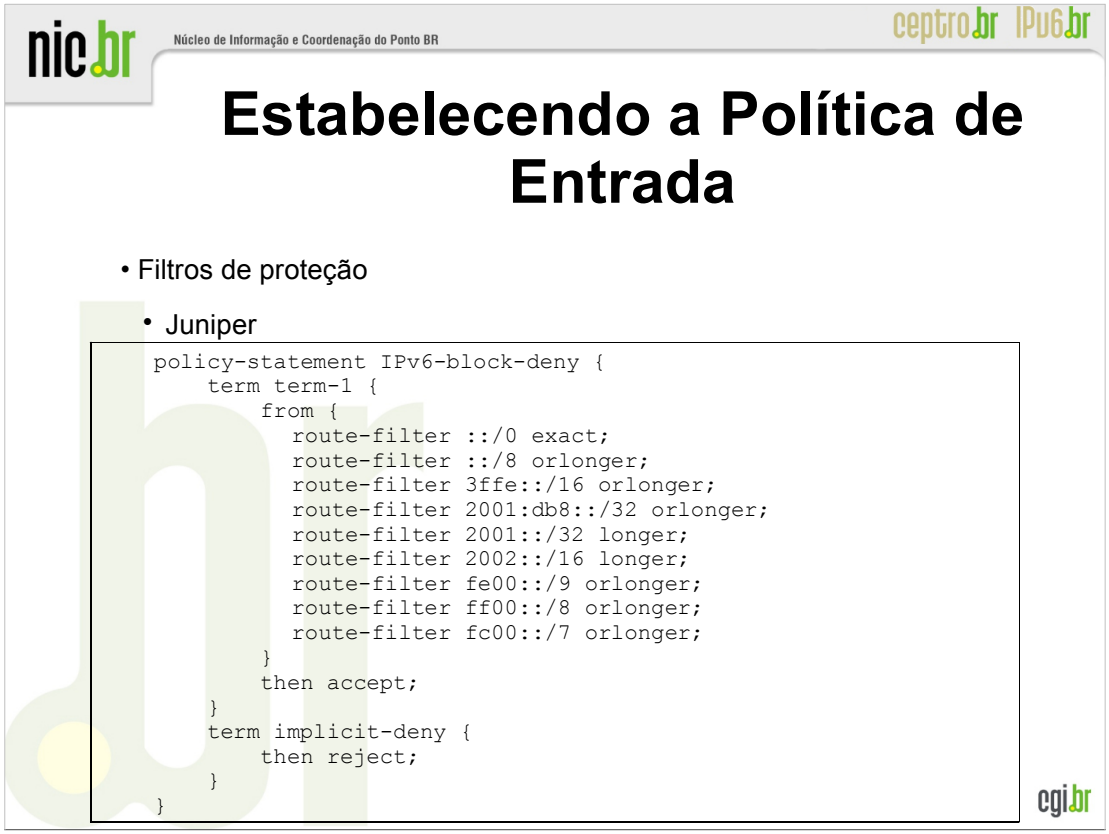
- A rota *default;*
- O primeiro bloco /8;
- Endereços da rede de testes 6bone;
- Endereços para documentação;
- Endereços dos túneis Teredo;
- Endereços dos túneis 6to4;
- Endereços *Link-local* (RFC 5735);
- Endereços Multicast;
- Endereços ULA.

Um exemplo da aplicação de um *prefix-list* de proteção para IPv4 poderia ser:

```
ip prefix-list IPv4-block-deny seq 10 permit 0.0.0.0/0
ip prefix-list IPv4-block-deny seq 20 permit 0.0.0.0/8
ip prefix-list IPv4-block-deny seq 30 permit 127.0.0.0/8
ip prefix-list IPv4-block-deny seq 40 permit 169.254.0.0/16
ip prefix-list IPv4-block-deny seq 50 permit 192.0.2.0/24
ip prefix-list IPv4-block-deny seq 60 permit 10.0.0.0/8
ip prefix-list IPv4-block-deny seq 60 permit 172.16.0.0/12
ip prefix-list IPv4-block-deny seq 80 permit 192.168.0.0/16
```

Estas *prefix-list* verificam respectivamente:

- A rota *default;*
- O primeiro bloco /8;
- O endereço de *loopback*;
- Endereços *Link-local* (RFC 5735);
- Endereços da TEST-NET-1 (RFC 5737);
- Endereços privados (RFC 1918).



Também é possível adicionar *route-fiter* de proteção. Observe o seguinte exemplo em um roteador Juniper:

```
route-filter ::/0 exact;
route-filter ::/8 orlonger;
route-filter 3ffe::/16 orlonger;
route-filter 2001:db8::/32 orlonger;
route-filter 2001::/32 longer;
route-filter 2002::/16 longer;
route-filter fe00::/9 orlonger;
route-filter ff00::/8 orlonger;
route-filter fc00::/7 orlonger;
```

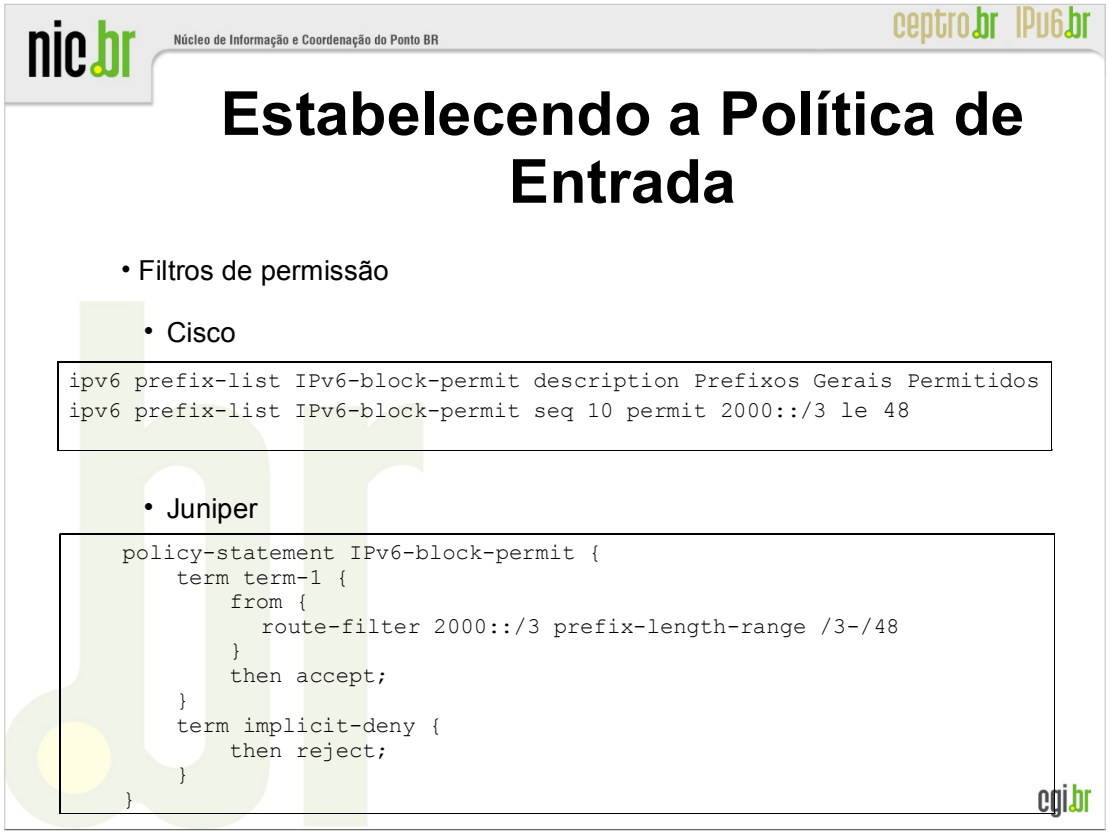
Estes *route-fiters* realizam as mesmas verificações apresentadas no exemplo anterior de roteadores Cisco.

Um exemplo da aplicação de um *prefix-list* de proteção para IPv4 poderia ser:

```
route-filter 0.0.0.0/0 exact;
route-filter 0.0.0.0/8 exact;
route-filter 127.0.0.0/8 exact;
route-filter 169.254.0.0/16 exact;
route-filter 192.0.2.0/24 exact;
route-filter 10.0.0.0/8 exact;
route-filter 172.16.0.0/12 exact;
route-filter 192.168.0.0/16 exact;
```

Mias informações:

- http://www.space.net/~gert/RIPE/ipv6-filters.html



Como vimos no módulo de Segurança IPv6, o modo de filtrarmos os endereços *bogons* no IPv6 é diferente da forma feita com IPv4. No IPv6 é mais fácil liberar as faixas de endereços alocados e bloquear o restante.

Os exemplos de `prefix-list` e `policy-statement` apresentados a seguir mostram uma forma flexível de liberar as faixas de endereços IPv6 disponíveis para alocação. Uma forma mais restrita de fazer esse mesmo tipo de filtro, é permitir uma a uma as faixas de endereços já atribuídas aos RIRs. Essas faixas podem ser obtidas em:

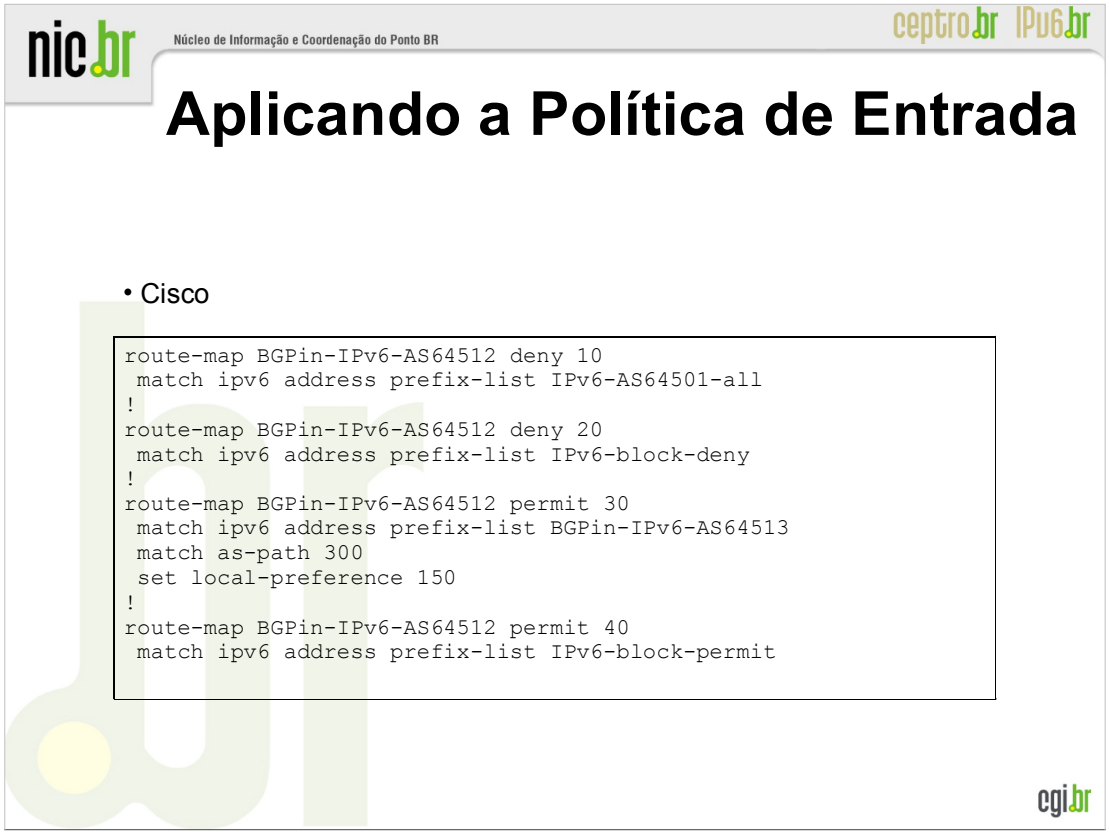
- http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml

Observe um exemplo de um roteador Cisco:

```
ipv6 prefix-list IPv6-block-permit description Prefixos Gerais Permitidos
ipv6 prefix-list IPv6-block-permit seq 10 permit 2000::/3 le 48
```

Observe um exemplo de um roteador Juniper:

```
policy-statement IPv6-block-permit {
    term term-1 {
        from {
               route-filter 2000::/3 prefix-length-range /3-/48
        }
        then accept;
    }
    term implicit-deny {
        then reject;
    }
}
```



Após serem estabelecidas as condições dos filtros via *prefix-list* nos roteadores Cisco, deve-se aplicá-las através de *route-maps*.

Ex.:

```
route-map BGPin-IPv6-AS64512 deny 10
 match ipv6 address prefix-list IPv6-AS64501-all
!
route-map BGPin-IPv6-AS64512 deny 20
 match ipv6 address prefix-list IPv6-block-deny
!
route-map BGPin-IPv6-AS64512 permit 30
 match ipv6 address prefix-list BGPin-IPv6-AS64513
 match as-path 300
 set local-preference 150
!
route-map BGPin-IPv6-AS64512 permit 40
 match ipv6 address prefix-list IPv6-block-permit
```

Esse é o *route-map* de entrada. Ele possui um nome para identificação de sua função, pois é possível ter vários *route-maps*. Neste exemplo, ele indica como tratar o que será recebido do AS 64512. Há 4 regras, a 10, 20, 30 e a 40.

Os *route-maps* trabalham com testes lógicos de “e” e “ou”, onde cada prefixo passa pelo *route-map*, e se a comparação entre a regra estabelecida e o prefixo coincidirem, ele é processado e a comparação é encerrada. Se a comparação não coincidir, a regra seguinte será analisada. Ou seja, cada regra é um teste “ou”.

A terceira regra possui dois testes de comparação. Quando há dois ou mais teste na mesma regra, é equivalente a condição “e”. Com isso, a terceira regra diz que tem que coincidir na primeira e na segunda linha.

O modo de funcionamento dos *route-map* é independente se a configuração é de uma sessão IPv4 ou IPv6.

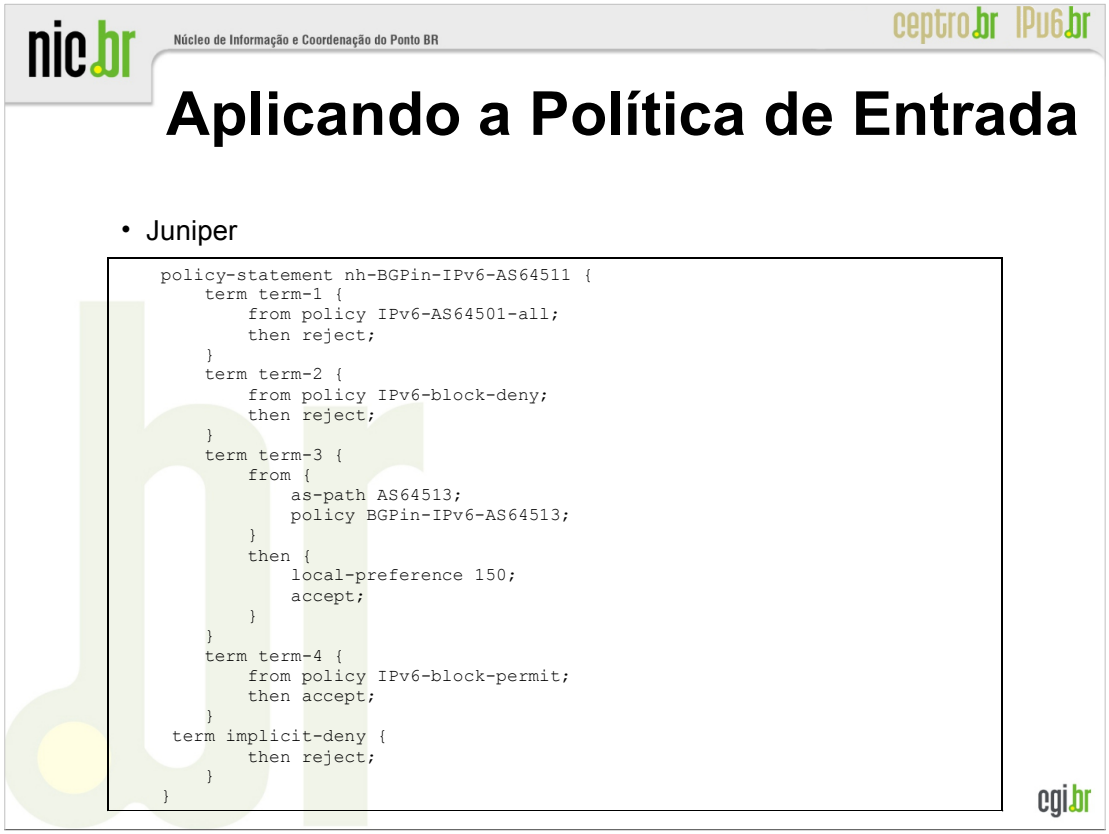
A primeira regra descarta todos os prefixos que coincidirem com o que foi estabelecido no `prefix-list IPv6-AS64501-all`, que representa todos os prefixos do próprio AS. É a regra que protege do sequestro de blocos.

A segunda regra descarta os blocos de uso privado especificados no `prefix-list IPv6-block-deny`.

A terceira regra é onde será alterado o *LOCAL_PREFERENCE*. Ela verifica de qual AS vem o prefixo (`as-path 300`) e se é o prefixo esperado (`prefix-list BGPin-IPv6-AS64513`). Se coincidir com as duas condições, o valor do *LOCAL_PREFERENCE* é aumentado para 150. O valor padrão do *LOCAL_PREFERENCE* é 100 e, quanto maior seu valor, maior a preferência.

A última regra, especificada no `prefix-list IPv6-block-permit`, permite o recebimento de anúncios de prefixos de dentro da faixa reservada pela IANA para alocação **2000::/3**. Ela permite anúncios de prefixos até um /48, tamanho normalmente aceito pelas operadoras.

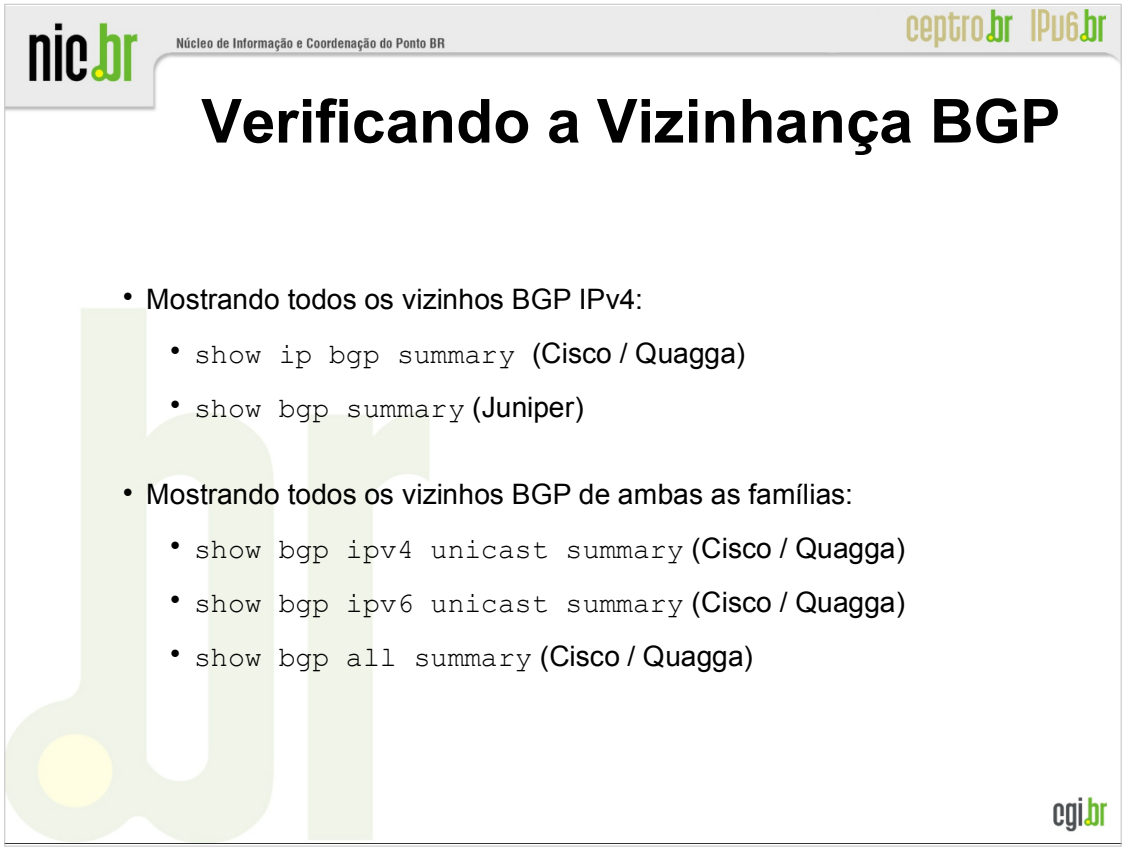
Qualquer outro anúncio recebido que não seja validado pelos *route-maps* serão descartados, pois os roteadores Cisco possuem um "*deny*" implícito como última regra.



A política implementada no roteador Juniper é similar a do roteador Cisco apresentada anteriormente. A principal diferença é que as políticas são aplicadas aos anúncios recebidos do AS 64511.

Ex.:

```
policy-statement nh-BGPin-IPv6-AS64511 {
    term term-1 {
        from policy IPv6-AS64501-all;
        then reject;
    }
    term term-2 {
        from policy IPv6-block-deny;
        then reject;
    }
    term term-3 {
        from {
            as-path AS64513;
            policy BGPin-IPv6-AS64513;
        }
        then {
            Local-preference 150;
            accept;
        }
    }
    term term-4 {
        from policy IPv6-block-permit;
        then accept;
    }
 term implicit-deny {
        then reject;
    }
}
```



Os comandos a seguir listam uma série de informações de estado da tabela BGP:

Mostrando todos os vizinhos BGP IPv4:

`show ip bgp summary` (Cisco / Quagga)

`show bgp summary` (Juniper)

Mostrando todos os vizinhos BGP de ambas as famílias:

`show bgp ipv4 unicast summary` (Cisco / Quagga)

`show bgp ipv6 unicast summary` (Cisco / Quagga)

`show bgp all summary` (Cisco / Quagga)

A partir dessas informações pode-se detectar uma série de problemas da sessão BGP.

nic.br Núcleo de Informação e Coordenação do Ponto BR ceptro.br IPv6.br

Verificando a Vizinhança BGP

- Cisco

```
router-R13#show bgp ipv6 unicast summary
BGP router identifier 172.21.15.253, local AS number 64501
BGP table version is 45, main routing table version 45
28 network entries using 4368 bytes of memory
54 path entries using 4104 bytes of memory
45/17 BGP path/bestpath attribute entries using 7560 bytes of memory
34 BGP AS-PATH entries using 848 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 2 (at peak 3) using 64 bytes of memory
BGP using 16944 total bytes of memory
26 received paths for inbound soft reconfiguration
BGP activity 49/1 prefixes, 96/21 paths, scan interval 60 secs

Neighbor          V      AS MsgRcvd MsgSent  TblVer InQ OutQ Up/Down State/PfxRcd
2001:DB8:21::254  4   64501    1867    1856       45   0    0 1w0d    Active
2001:DB8:21::255  4   64501    4136    3642       45   0    0 1d07h           26
2001:DB8:20::255  4   64512    1896    1876       45   0    0 1d07h            0
```

egi.br

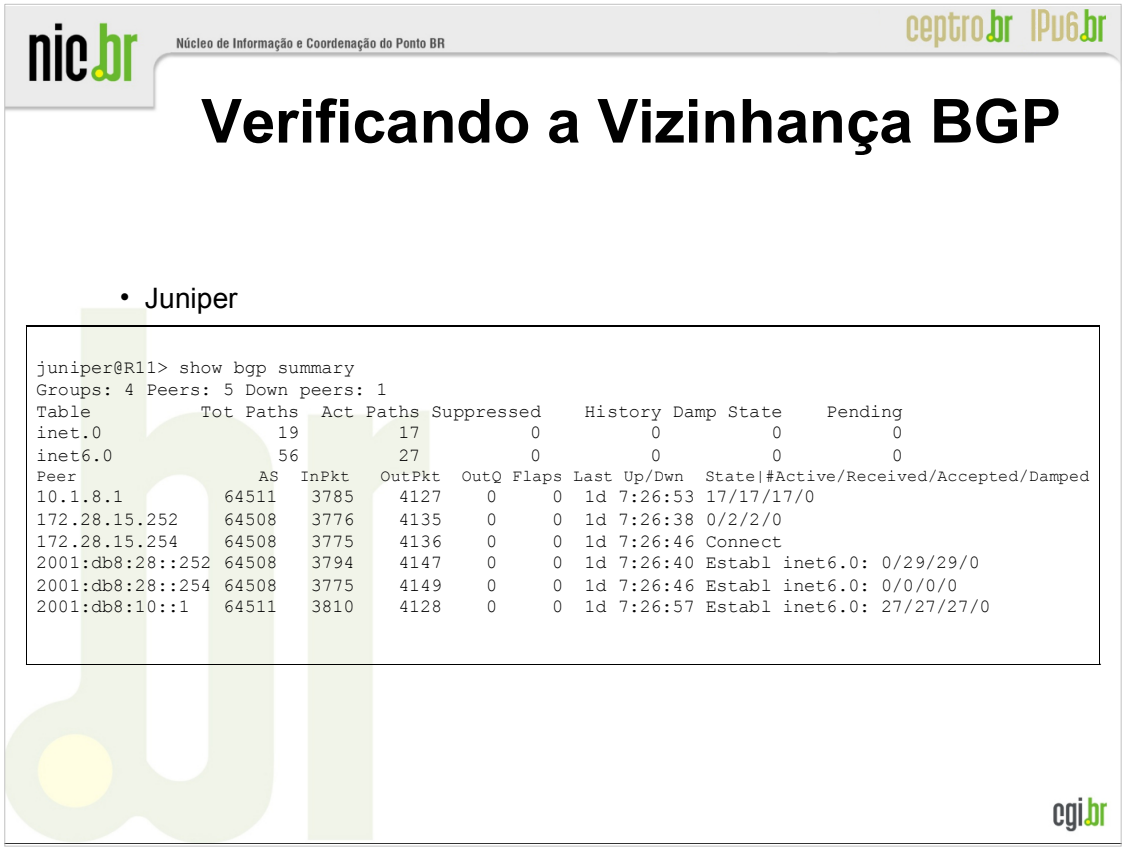
Neste exemplo, podemos observar os seguintes dados sobre os vizinhos das sessões BGP IPv6 em um roteador Cisco:

- Neighbor – IP do vizinho em que se estabeleceu a sessão BGP;
- V – versão do BGP;
- AS – ASN do vizinho;
- MsgRcvd – quantidade de mensagens recebidas do vizinho;
- MsgSent – quantidade de mensagens enviadas ao vizinho;
  - esses dois últimos campos normalmente não são muito verificados, mas são importantes. Muita variação desses campos pode identificar um problema. Receber muitas mensagens, se a tabela está sendo atualizada muitas vezes, pode indicar uma flutuação grande com seu vizinho.
- TblVer – versão da tabela;
- InQ – fila de entrada de pacotes;
- OutQ – fila de saída de pacotes;
- Up/Down - tempo da última mudança de estado;
- State/PfxRcd – indica o estado atual ou o número de prefixo aprendidos. Lembre-se, apesar de existirem estados como Active e Established, que aparentemente indicam que a sessão está ok, eles apenas representam estados intermediários da conexão. A sessão só estará plenamente estabelecida quando houver a indicação de quantos prefixos foram aprendidos.

Observe a saída do mesmo comando, mas agora para visualizar as informações sobre a vizinhança BGP IPv4 em um roteador Cisco:

```
router-R13#show bgp ipv4 unicast summary
BGP router identifier 172.21.15.253, local AS number 64501
BGP table version is 80, main routing table version 80
31 network entries using 4092 bytes of memory
52 path entries using 2704 bytes of memory
33/22 BGP path/bestpath attribute entries using 5544 bytes of memory
28 BGP AS-PATH entries using 672 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 3 (at peak 4) using 96 bytes of memory
BGP using 13108 total bytes of memory
21 received paths for inbound soft reconfiguration
BGP activity 99/40 prefixes, 185/84 paths, scan interval 60 secs

Neighbor        V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down
State/PfxRcd
10.2.255.255    4   64512   10578   10474       80    0    0 1w0d     Active
172.21.15.254   4   64501   10544   10490       80    0    0 1w0d           0
172.21.15.255   4   64501   10572   10490       80    0    0 1w0d          21
```



Neste exemplo, podemos observar os seguintes dados sobre os vizinhos das sessões BGP em um roteador Juniper:

- Groups - número de grupos BGP;
- Peers – números de vizinhos BGP;
- Down peers – número de vizinhos BGP desconectados;
- Table – nome da tabela de rotas;
- Tot Paths – número total de caminhos;
- Act Paths – número de rotas ativas;
- Suppressed - número de rotas atualmente inativas. Estas rotas não aparecem na tabela de encaminhamento e não são exportadas pelos protocolos de roteamento;
- History - número de rotas retiradas armazenadas localmente para manter o controle histórico de instabilidade;
- Damp State – número de rotas com uma figura de mérito maior que zero, mas que continuam ativas porque o valor não atingiu o limite em que a retirada ocorre;
- Pending – rotas em processamento pela política de importação do BGP;
- Peer – endereço de cada vizinho BGP;
- AS – ASN do vizinho;
- InPkt – número de pacotes recebidos do vizinho;
- OutPkt – númrero de pacotes enviados ao vizinho;
- OutQ - fila de saída de pacotes;
- Flaps – número de vezes que a sessão BGP foi interrompida e se re-estabeleceu;
- Last Up/Down – última vez em que ocorreu uma mudança de estado;
- State|#Active/Received/Accepted/Damped - indica o estado atual ou o número de prefixo aprendidos. Se a sessão não foi estabelecida, este campo mostra o estado atual da sessão: Active, Connect, ou Idle. Se a sessão foi estabelecida, o campo indica o número de rotas ativas, recebidas, aceitas ou instáveis.

Observe que o Juniper apresenta na saída do mesmo comando as informações sobre as sessões IPv4 e IPv6.

É importante verificar através de *Looking Glasses* remotos como as operadoras e toda a Internet recebem os anúncios do AS. Deste modo, é possível verificar também, se suas políticas de roteamento foram bem aplicadas.

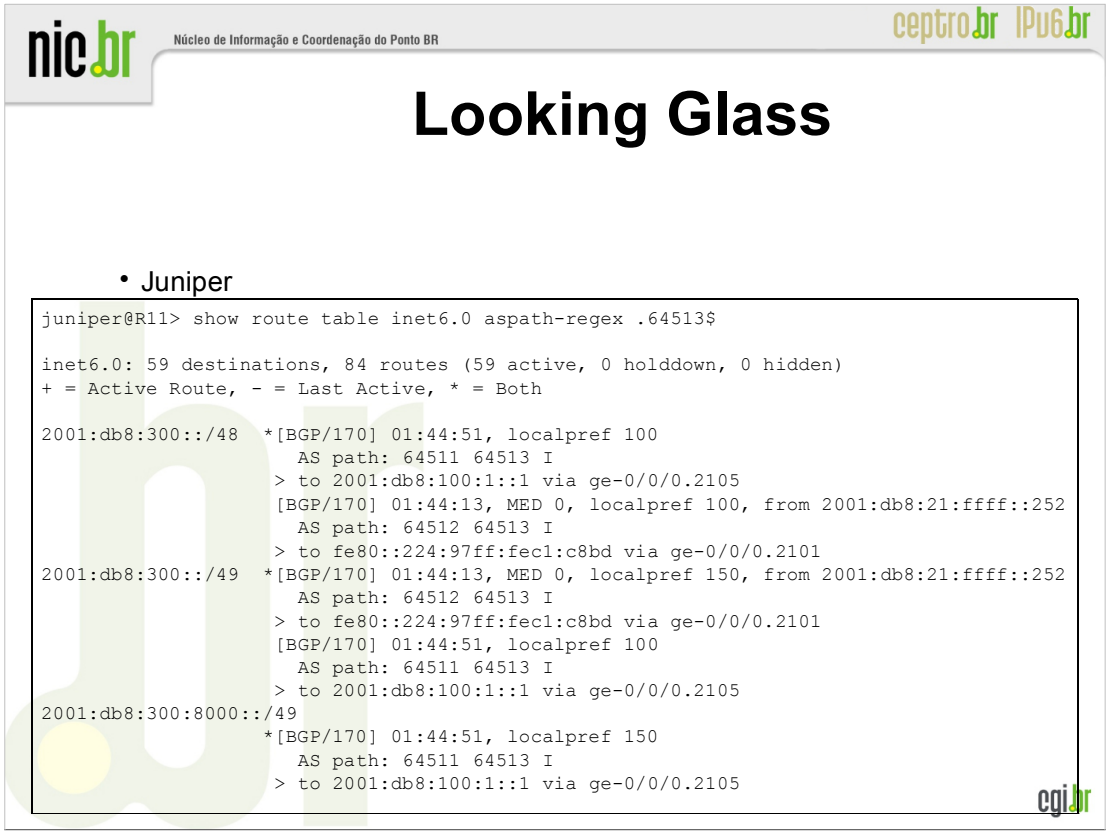
Com o *Looking Glass*, é possível consultar como os prefixos de um AS, IPv4 e IPv6, estão sendo aprendidos pela Internet, ou seja, como os ASs conseguem chegar até a sua rede.

Ex.:

`show bgp regexp _64501$` (Cisco / Quagga)

Neste exemplo, em um roteador Cisco, podemos observar como cada prefixo anunciado pelo AS 64501 foi aprendido. Nesta expressão regular, o "$" significa que é a origem (início de linha), e o "_" significa espaço, ou seja, o comando é aplicado para todo prefixo que tem o AS 64501 como origem no AS-PATH.

Na resposta a consulta podemos observar o balanceamento do tráfego, pois o bloco /48 foi dividido em dois prefixos /49 permitindo que metade do tráfego saia por um *link* e a outra metade saia por outro *link*, e também a redundância das rotas, pois além dos prefixos /49 o prefixo /48 também está sendo anunciado pelos dois *links*.



Neste exemplo podemos observar, em um roteador Juniper, como os prefixo anunciado pelo AS64513 foram aprendidos pelo AS64501. A expressão regular utilizada é similar a vista no exemplo anterior utilizando roteadores Cisco.

Na resposta a consulta podemos observar o balanceamento do tráfego e também a redundância das rotas, assim como no exemplo anterior, além de informações como: melhor rota, indicado pelo asterisco; caminho até o destino (*AS path*); próximo salto; e interface de saída do pacote.

Mias informações:

• Routing TCP/IP
Autor: Jeff Doyle,Jennifer DeHaven Carroll

• O Protocolo BGP4 - Parte 3 (Final) - http://www.rnp.br/newsgen/9907/pgbp4p3.html
Autor: RNP – Rede Nacional de Ensino e Pesquisa