CURSO AVANÇADO DE BGP DESIGN COM ROTEADORES CISCO

- Instrutor: Rinaldo Vaz
- Analista de Redes
- Responsável técnico pelo AS 28135

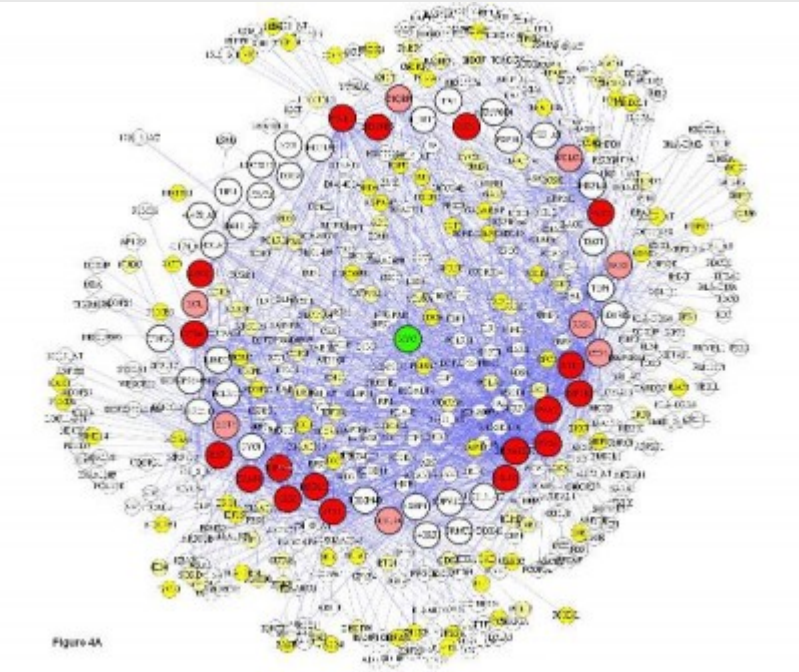
Versão do material: 1.1
Última publicação: 23/04/2012

1.1
INTRODUÇÃO

1.1 Introdução

- A versão corrente do BGP é a versão 4, especificada na RFC 1771

- É o protocolo responsavel por fazer a internet funcionar da maneira que é



1.1 Introdução

- Algumas Características:

1. É um protocolo de vetor caminho;
2. As tabelas completas de roteamento são trocadas entre os doisa routers no início da sessão e as atualizações adicionais incrementais são enviadas em seguida;
3. Atualizações adicionais são enviadas imediatamente através de mensagens de update;
4. Utiliza por padrão a porta TCP 179;

1.1 Introdução

- Mensagens BGP:

1. Open (abertura) - é a primeira mensagem enviada por um router que deseja estabelecer uma sessão com outro, esse por sua vez também envia uma mensagem "open", após recebidas essas mensagens, cada router envia a primeira mensagem de "keep alive" e a sessão é estabelecida

2. Update (atualização) - É dentro dessas mensagens que vão informações sobre cada prefixo que está sendo anunciado, uma mensagem de update também pode conter informações sobre redes que ficaram indisponíveis e devem ser removidas

3. Notificação (notification message) – é enviada quando uma condição de erro é detectada; elas são usadas para encerrar uma sessão ativa e informar a quaisquer roteadores conectados do porque do encerramento da sessão.

4. Keep-alive (ainda estou aqui) - de tempo em tempo cada router envia uma mensagem de keep-alive para que o vizinho saiba que há conectividade IP. Caso o keep-alive atrase, o router começa a contagem de "hold-time" e se nesse período não for recebido nenhum keep-alive a sessão é finalizada. Tanto o tempo de keep-alive como hold-time podem ser configurados para mais ou menos tempo de acordo com o tipo de link.

1.1 Introdução

- Diferenças entre FIB e RIB

Fowarding Information Base (FIB)

- Quando um router recebe 2 rotas diferentes para o mesmo prefixo os critérios de escolha de rota são analizados. A rota que foi considerada melhor de acordo com os critérios BGP fica na FIB

Router information base (RIB)

- As rotas que não foram consideradas melhores ficarão guardadas na RIB e utilizadas caso a rota principal fique indisponível
- O BGP jamais anuncia uma prefixo que não esteja na FIB

1.1 Introdução

- Distancias administrativas:

Outro ponto importante é a distância administrativa, caso 2 prefixos sejam idênticos, o router analisa sua distância administrativa, a menor sempre "ganha". Caso sejam iguais os criterios particulares do protocolo de roteamento correspondente serão analizados

Directamente Conectada 0
Estática 1
eBGP 20
EIGRP (Interno) 90
IGRP 100
OSPF 110
ISIS 115
RIP 120
EGP 140
EIGRP (Externo) 170
iBGP 200
BGP Local 200
Desconhecido 255

1.2
REQUISITOS PARA SE TORNAR UM SISTEMA AUTÔNOMO

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

1.2 - Requisitos para se tornar um sistema autônomo

- http://registro.br/provedor/numeracao/regras.html
- Uma organização justifica a designação de um ASN quando apresenta uma das seguintes necessidades:
  - Multi Provedor: Quando a organização está conectada a dois ou mais provedores de transito Internet distintos e independentes e necessita, portanto, fazer uso de protocolos de roteamento dinâmico.
  - Política única de roteamento: Quando a organização possui uma política de roteamento que é distinta daquela aplicada pelo(s) provedor(es) de transito Internet.

1.3
PREENCHENDO O FORMULÁRIO DO REGISTRO.BR

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

1.3 - Preenchendo o formulário do Registro.br

- Há um formulário simples que pode ser baixado no seguinte endereço:

  http://registro.br/provedor/numeracao/pedido-form.txt

- Em caso de dúvidas há um arquivo contento informações de ajuda:

  http://registro.br/provedor/numeracao/pedido-ajuda.txt

- Informações completas em:

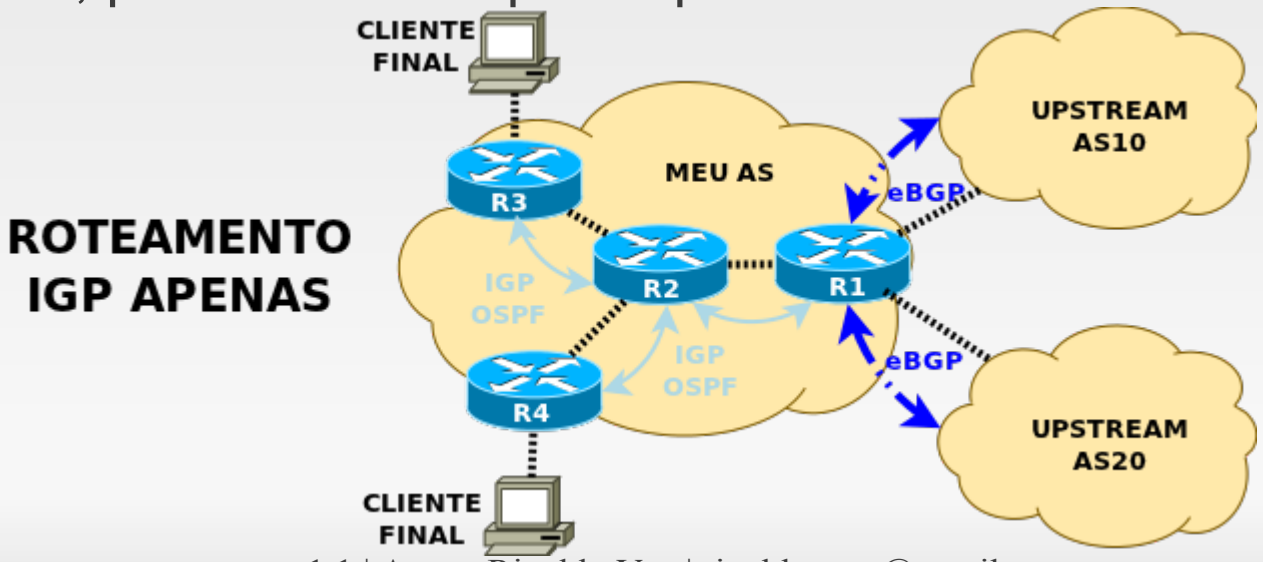
  http://registro.br/provedor/numeracao/faq.html

1.4
QUANDO UTILIZAR IGP E EGP

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

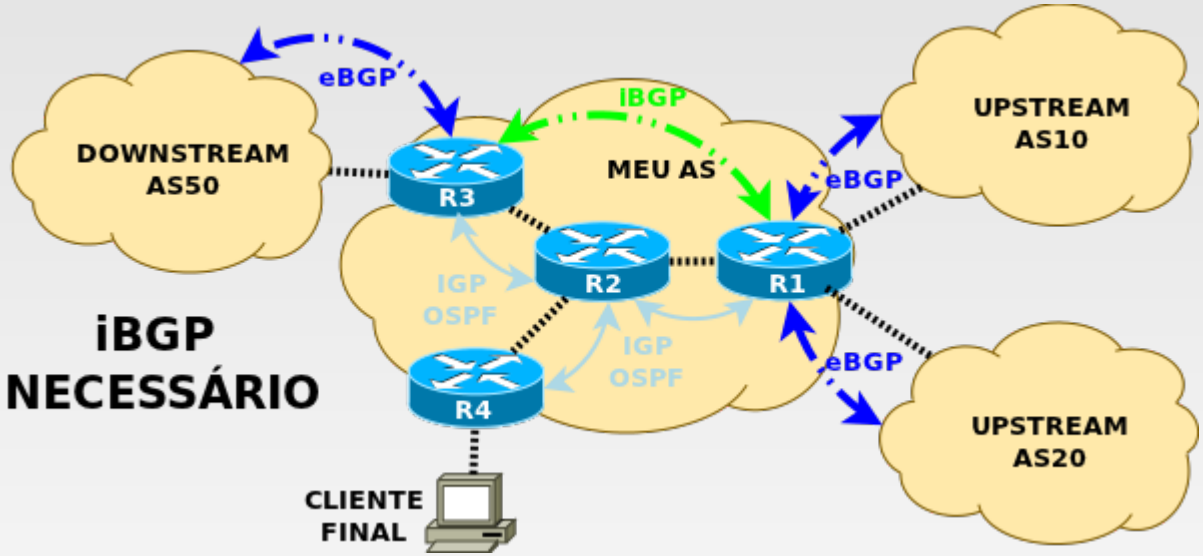
1.4 - Quando utilizar IGP e EGP

- O BGP foi desenvolvido para troca de rotas entre sistemas autônomos distintos, sendo assim, não há nenhum motivo para um provedor de serviços utilizar BGP se esse não possuir um ASN
- A finalidade do BGP é permitir que os IPs do sistema autônomo apareçam na internet
- Tecnicamente nada impede de utilizar BGP com apenas uma operadora, porém é um requisito para obter um ASN



1.4 - Quando utilizar IGP e EGP

- Caso um dos clientes do provedor também seja um sistema autônomo é indispensável a utilização do BGP com o cliente e operadoras de trânsito



1.5
ENTENDENDO O PROCESSO DE ESCOLHA DE ROTAS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

1.5 - Entendendo o processo de escolha de rotas

- Em primeiro lugar, o BGP apenas analisa os atributos para "desempatar" duas ou mais rotas para o **MESMO PREFIXO**

- Caso haja um prefixo mais específico esse sempre será preferido independentemente dos seus atributos "mais favoráveis"

1.5 - Entendendo o processo de escolha de rotas

- Como o BGP escolhe uma rota:
- Sempre que houverem 2 ou mais opções de rota para prefixos iguais (mesma máscara de subrede) e ambos tiverem sido recebidos via BGP o protocolo vai escolher a melhor rota de acordo com a ordem:

1. *Rota com maior valor de WEIGHT*
2. *Rota com maior valor de LOCAL_PREF*
3. *Rota originada localmente (bgp network)*
4. *Rota com o menor AS_PATH.*
5. *Rota com menor tipo de origem.*
   IGP (i) < EGP (e) < INCOMPLETE (?)
6. *Rota com menor métrica multi-exit discriminator (MED).*
7. *Escolhe a rota recebida por (eBGP) em relação a uma (iBGP).*

1.5 - Entendendo o processo de escolha de rotas

- Como o BGP escolhe uma rota:

  8. Rota com a menor métrica IGP para o nexthop BGP
  ex: o next-hop aprendido via OSPF vai "vencer" um next-hop aprendido via ISIS

  9. Rota externa mais antiga

  10. Rota recebida de um router com menor Router ID

  11. Rota com o menor tamanho de cluster list. Ambientes com Route Reflector apenas

  12. Rota com o menor endereço de neighbor

- A análise segue para o próximo critério apenas quando há empate no critério atual, assim, o tamanho do AS PATH só será analisado caso o valor de weight e local preference sejam os mesmos para as duas rotas

2.1
CONFIGURANDO UMA SESSÃO BGP COM MINHA OPERADORA

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

2.1 Configurando uma sessão BGP com minha operadora

- O router de cada grupo responde pelo IP **X**.128.0.1/9 (onde **X** é o número do grupo)
- Para acessa-lo, configurem nos seus laptops um IP dentro dessa mesma rede preferencialmente .2 e gateway padrão **X**.128.0.1
- Verifiquem a conectividade IP e em caso positivo abram uma sessão telnet com esse equipamento:

```
ALUNO@notebook# telnet X.128.0.1
Trying X.128.0.1...
Connected to X.128.0.1.
Escape character is '^]'.
User Access Verification

Password:
```

A senha de acesso é 123

2.1 Configurando uma sessão BGP com minha operadora

- Estamos agora no modo ”usuário”:

  GX-R1>

- Para iniciar as configurações precisaremos inicialmente entrar no modo ”privilegiado”, e será exigida uma outra senha:

  G1-R1>enable

  Password: gXr1

  GX-R1#

- E em seguida no modo de configuração ”global”

  GX-R1#configure terminal

  Enter configuration commands, one per line. End with CNTL/Z.

  GX-R1(config)#

2.1 Configurando uma sessão BGP com minha operadora

- O primeiro passo na configuração BGP é adicionar um neighbor (peer)

```
GX-R1(config)# router bgp X
GX-R1(config-router)#
        neighbor 100.X.1.1 remote-as 100
        neighbor 100.X.1.1 description eBGP com as 100
```

2.1 Configurando uma sessão BGP com minha operadora

- Verificando o status da sessão BGP recém criada

GX-R1#sh ip bgp summary

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	7	4	4	0	0	00:00:02	2

2.1 Configurando uma sessão BGP com minha operadora

- Verificando os anúncio que recebo da minha operadora

GX-R1#show ip bgp neighbors 100.X.1.1 routes

Network	Next Hop	Metric	LocPrf	Weight	Path
...					
*> 100.0.0.0	100.X.1.1			0	100 i
*> 123.123.123.0/24	100.X.1.1			0	100 111 55 123 i
…					

- Posteriormente veremos na prática a diferença entre os comandos com ”received-routes” e apenas ”routes” no final

2.1 Configurando uma sessão BGP com minha operadora

- Agora todos tentam ping para 123.123.123.1 a partir do seu router:

  Sucesso??
- Agora um traceroute

- Em seguida ping para 123.123.123.1 a partir do seu laptop

  Sucesso??
- Agora um Traceroute

- Que problema pode estar havendo?

2.1 Configurando uma sessão BGP com minha operadora

- Esse ponto merece uma pausa para analisar algo importante no ”Troubleshooting”

- Ao executar um traceroute um host verifica a sua tabela de roteamento, caso não haja ocorrências os pacotes do traceroute são encaminhados pela rota padrão, não sendo nada especificado no comando, o IP de origem é sempre o mesmo da interface conectada ao ”next-hop”

- Em nosso LAB qual o next-hop para 123.123.123.1 ?

  provavelmente 100.**X**.1.1, qual IP o router vai utilizar como origem?

2.1 Configurando uma sessão BGP com minha operadora

- O IP de origem é 100.**X**.1.2 que por sua vez está dentro de uma rede ”conhecida” pelo looking glass (123.123.123.1)

- Já o laptop não consegue alcançar o destino porque o seu IP de origem é **X**.128.0.2, rede que não é conhecida pelo looking glass já que ainda não anunciamos nada

- Não se trata então de o laptop ”não conseguir alcançar” 123.123.123.1, na verdade os pings disparados pelo laptop chegam ao seu destino, porém o destino não sabe para quem responder

2.1 Configurando uma sessão BGP com minha operadora

- O segundo passo é anunciar o classe A (bloco /8) do seu grupo

```
GX-R1# router bgp X
!--- entra no nodo de configuração global
GX-R1(config)# router bgp X
!--- entra no nodo de configuração BGP
GX-R1(config-router)#network X.0.0.0
!--- adiciona a rede x.0.0.0 para ser anunciada a partir desse router
GX-R1(config-router)#end
!--- volta ao modo provilegiado
GX-R1# clear ip bgp 100.X.1.1
!--- reinicia a sesão BGP
```

2.1 Configurando uma sessão BGP com minha operadora

- Agora verifico os anúncios que estou fazendo para o AS 100

GX-R1#sh ip bgp neighbors 100.X.1.1 advertised-routes

BGP table version is 15, local router ID is 100.X.1.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> X.0.1.0/24	0.0.0.0	0		32768	i

- Caso apareça algo parecido com isso, vamos testar novamente a conectividade IP com o looking glass a partir do laptop

2.1 Configurando uma sessão BGP com minha operadora

- Sintam-se à vontade para entrar no looking glass e executar testes de PING e TRACEROUTE

2.2
ENTENDENDO ATRIBUTOS NAS ROTAS RECEBIDAS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar melhor as rotas recebidas

  GX-R1>sh ip bgp summary

  ...

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- Esse comando mostra todas as sessões BGP configuradas ativas e não ativas.
- Nesse laboratório apenas uma sessão está configurada
- Vamos entender o que cada campo significa...

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **Neighbor = 100.X.1.1**
- Esse é o IP do router remoto (configurado em router BGP **X**)

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **V = 4**
- Versão do BGP utilizada nessa sessão

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **AS = 100**
- AS do neighbor remoto (configurado em neighbor 100.x.1.1 remote-as X)

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **MsgRcvd = 112**
- **MsgSent = 83**
- Quantidade total de mensagens BGP recebidas e enviadas incluindo ”keepalives” e ”updates”

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **TblVer = 22**
- Versão atual da tabela de rotas BGP, sempre que uma nova rota é inserida ou removida o **TblVer** é incrementado

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **InQ = 0**
- **OutQ = 0**
- Updates na fila de entrada e de saída

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **Up/Down = 01:06:03**
- Contador das horas minutos e segundos desde de que a sessão foi estabelecida ou que a sessão ficou down, após 24 horas o contador muda para **1d00h00m** (dias/horas/minutos) após uma semana **01w0d00h (**semanas/dias/horas)
- Antes de uma sessão estabelecer ao menos uma vez o contador fica em ”never”

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.X.1.1	4	100	112	83	22	0	0	01:06:03	3

- **State/PfxRcd = 3**
- Esse campo merece uma atenção especial pois será o mais frequentemente observado. Indica o estado da conexão BGP
  - Nesse caso está contando 3 prefixos, mas existem 5 estados possíveis, **IDLE, CONECT, ACTIVE, OPENSENT, OPENCONFIRM e ESTABLISHED,** iremos entender melhor cada um desses estados nos próximos slides

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

State/PfxRcd

IDLE

- É o estado em que a sessão fica quando é interrompida por uma mensagem de notificação ou intervenção do administrador, esse estado durante muito tempo aponta algum possível problema de conectividade IP ou ainda que o BGP não foi configurado corretamente do lado remoto

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

State/PfxRcd

ACTIVE

- ACTIVE: O BGP **tenta estabelecer** comunicação com um peer inicializando uma conexão TCP. Caso esta seja bem sucedida, passa-se ao estado OPENSENT. Se esta tentativa não for bem sucedida, pelo motivo de expiração do tempo, por exemplo, o estado passa para CONNECT. Em cada interrupção pelo sistema ou pelo administrador, volta ao estado IDLE. Geralmente as transições entre o estado de CONNECT e ACTIVE refletem problemas no nível TCP.

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

State/PfxRcd

CONNECT

- Neste estado o BGP **aguarda** pela conexão TCP, com destino na porta 179. Quando a conexão estiver estabelecida, ou seja, com o recebimento da mensagem de OPEN, passa-se ao estado de OPENSENT. Se a conexão nível de transporte não for bem sucedida, o estado vai para ACTIVE. No caso do tempo de espera ter sido ultrapassado, o estado volta para CONNECT. Em qualquer outro evento o estado retorna para IDLE.

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

State/PfxRcd

OPEN SENT

- OPENSENT: Neste estado o BGP aguarda pela mensagem de OPEN e faz uma checagem de seu conteúdo. Caso seja encontrado algum erro como número de AS incoerente ao esperado ou a própria versão do BGP, envia-se uma mensagem tipo NOTIFICATION e volta ao estado de IDLE. Caso não ocorram erros na checagem, inicia-se o envio de mensagens KEEPALIVE. Em seguida, acerta-se o tempo de Hold Time, sendo optado o menor tempo entre os dois peers. Depois deste acerto, compara-se o número AS local e o número AS enviado pelo peer, com o intuito de detectar se é uma conexão iBGP (números de AS iguais) ou eBGP (números de AS diferentes). Em caso de desconexão a nível TCP, o estado passa para ACTIVE. Para as demais situações de erro, como expiração do Hold Time, envia-se uma mensagem de NOTIFICATION com o código de erro correspondente e retorna-se ao estado de IDLE. No caso de intervenção do administrador ou o próprio sistema, também retorna-se o estado IDLE.

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

State/PfxRcd

OPEN CONFIRM

- OPENCONFIRM: Neste estado o BGP aguarda pela mensagem de KEEPALIVE e quando esta for recebida, o estado segue para ESTABLISHED e a negociação do peer é finalmente concluída. Com o recebimento da mensagem de KEEPALIVE, é acertado o valor negociado de Hold Time entre os peers. Se o sistema receber uma mensagem tipo NOTIFICATION, retorna-se ao estado de IDLE. O sistema também envia periodicamente, segundo o tempo negociado, mensagens de KEEPALIVE. No caso da ocorrência de eventos como desconexão ou intervenção do operador, retorna-se ao estado de IDLE também. Por fim, na ocorrência de eventos diferentes aos citados, envia-se uma mensagem NOTIFICATION, retornando ao estado de IDLE.

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

State/PfxRcd

ESTABLISHED

- ESTABLISHED: Neste estado, o BGP inicia a troca de mensagens de UPDATE ou KEEPALIVE, de acordo com o Hold Time negociado. Caso seja recebida alguma mensagem tipo NOTIFICATION, retorna-se ao estado IDLE. No recebimento de cada mensagem tipo UPDATE, aplica-se uma checagem por atributos incorretos, inconsistentes, duplicados, ou mesmo faltando, caso algum erro seja detectado, envia-se uma mensagem de NOTIFICATION, retornando ao estado IDLE. Por fim, se o Hold Time expirar ou for detectada desconexão ou intervenção do administrador, também retorna-se ao estado de IDLE.

2.2 Entendendo atributos nas rotas recebidas

- Entendendo os campos do summary:

State/PfxRcd

3

- Esse número não representa a quantidade de prefixos recebidos de um neighbor, mas sim a quantidade de prefixos que foram parar na RIB após terem passado pelos filtros
- Um State/PfxRcd 0 não significa necessariamente que não estou recebendo nenhuma rota, por exemplo, pode ser que o next-hop dessas rotas seja inalcançável pelo meu router e assim elas não aparecem. Detalharemos esses casos posteriormente.

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar todas as rotas BGP

  GX-R1# show ip bgp

  BGP table version is 22, local router ID is 100.x.1.2

  Status codes: s suppressed, d damped, h history, * valid, > best, i - internal

  Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric LocPrf Weight Path
*> 6.0.0.0	100.X.1.1	0 100 6 i
*> 100.0.0.0	100.X.1.1	0 100 i
*> 123.0.0.0	100.X.1.1	0 100 222 123 i

- **Esse comando mostra todas recebidas via BGP escolhidas (FIB) e não escolhidas (RIB). Como recebemos rotas de apenas um peer, não há rotas "não escolhidas"**

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar uma rota específica:

GX-R1#sh ip bgp 123.0.0.0

BGP routing table entry for 123.0.0.0/8, version 10

Paths: (1 available, best #1, table Default-IP-Routing-Table)

Flag: 0x208

Not advertised to any peer

100 222 123

100.X.1.1 from 100.X.1.1 (100.100.100.1)

Origin IGP, localpref 100, valid, external, best

#Esse comando detalha as rotas para o prefixo 123.0.0.0/8, e caso outras rotas para esse mesmo prefixo tenham sido recebida via BGP de outro(s) neighbor(s), mais entradas semelhantes aparecerão uma abaixo da outra

- **Vamos entender cada campo a seguir:**

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar uma rota em específico

  GX-R1#sh ip bgp 123.0.0.0

  BGP routing table entry for 123.0.0.0/8, version 22

  ...

- **Essa linha mostra a versão da routing table descrita nos slides anteriores**

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar uma rota em específico

  GX-R1#sh ip bgp 123.0.0.0

  ...

  Paths: (1 available, best #1, table Default-IP-Routing-Table)

- Paths significa a quantidade de neighbors que me enviaram rotas para esse mesmo prefixo, ou seja, ”1 avaliable” diz que apenas 1 neighbor me enviou.
- ”best #1” a rota que foi escolhida (na ordem de cima para baixo) de acordo com os critérios do BGP – weigh, local_pref, etc
- ”table Default-IP-Routing-Table” significa que as rotas estão na FIB padrão. Múltiplas FIB's são usadas com VRF

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar uma rota em específico

  GX-R1#sh ip bgp 123.0.0.0

  ...

  Flag: 0x208

  ...

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar uma rota em específico

  GX-R1#sh ip bgp 123.0.0.0

  ...

  Not advertised to any peer

  ...

- **Essa mensagem significa que esse prefixo 123.0.0.0/8 não está sendo anunciada via BGP para nenhum neighbor. Na verdade qualquer rota nesse laboratório aparecerá com essa mensagem.**
- **Esse campo será detalhado posteriormente na sessão ”AS de trânsito”**

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar uma rota em específico

  GX-R1#sh ip bgp 123.0.0.0

  ...

  100 111 55 123

  ...

- AS-PATH: mostra da direita para a esquerda o AS que originou essa rota e todos os ASs no caminho
- Ou seja, o AS 123 originou o anúncio para o AS 222 que recebeu, considerou como melhor rota e exportou para o as 100 o qual também considerou como melhor e repassou para meu AS
- Um ponto importante a se observar é que mesmo que um AS tenha mil caminhos para um determinado prefixo, é apenas UM que ele vai escolher como ”BEST #” e passar adiante

2.2 Entendendo atributos nas rotas recebidas

- Vamos detalhar uma rota em específico

  GX-R1#sh ip bgp 123.0.0.0

  ...

  100.X.1.1 from 100.X.1.1 (x.x.x.x)

  ...

- O primeiro IP dessa linha significa o ”next-hop”(próximo salto) para esse prefixo, caso o next-hop não esteja contido em uma rota válida da FIB esse jamais será eleito como ”BEST”
- O segundo IP dessa linha me diz o endereço do ”neighbor” que me enviou essa rota via BGP
- O IP (X.X.X.X) é na verdade o ”router-ID” desse neigbor descrito acima

2.3
Configurando conexões BGP com duas operadoras (multihoming)

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

2.3 Configurando conexões BGP com duas operadoras

- Primeiro passo é a configuração do próximo neighbor

```
GX-R1(config)# router bgp X
GX-R1(config-router)#
        network x.0.0.0 mask 255.0.0.0
        neighbor 100.x.1.1 remote-as 100
        neighbor 100.x.1.1 description eBGP com as 100
        !--- O neighbor acima foi já configurado
        !--- próximo Neigbor
        neighbor 200.X.1.1 remote-as 200
        neighbor 200.X.1.1 description eBGP com as 200
```

2.3 Configurando conexões BGP com duas operadoras

- Verificando o status das sessões BGP recém criadas

GX-R1#sh ip bgp summary

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
100.1.1.1	4	100	192	170	9	0	0	00:27:48	7
200.1.1.1	4	200	191	181	9	0	0	00:27:51	7

- O número de PfxRcd vai ser maior a medida que os alunos estabeleceram suas conexões

2.3 Configurando conexões BGP com duas operadoras

- Detalhando as rotas recebidas via BGP
- Em 23 dos 24 routers parte da saída desse comando aparecerá assim:

G1-R1>sh ip bgp
BGP table version is 9, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	100.1.1.1			0	100 2 i
*	200.1.1.1			0	200 2 i
*> 55.0.0.0	100.1.1.1			0	100 111 55 i
*	200.1.1.1			0	200 222 123 55 i
*> 100.0.0.0	100.1.1.1			0	100 i
*	200.1.1.1			0	200 2 100 i
*> 111.0.0.0	100.1.1.1			0	100 111 i
*	200.1.1.1			0	200 2 100 111 i
* 123.0.0.0	100.1.1.1			0	100 111 55 123 i
*>	200.1.1.1			0	200 222 123 i
* 200.0.0.0/8	100.1.1.1			0	100 2 200 i
*>	200.1.1.1			0	200 i
* 222.0.0.0/8	100.1.1.1			0	100 2 200 222 i
*>	200.1.1.1			0	200 222 i

2.3 Configurando conexões BGP com duas operadoras

- Detalhando as rotas recebidas via BGP
- Porém em 1 dos 24 routers a saída será direfente:

G1-R1>sh ip bgp
BGP table version is 9, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	100.1.1.1			0	100 2 i
*	200.1.1.1			0	200 2 i
*> 55.0.0.0	100.1.1.1			0	100 111 55 i
*	200.1.1.1			0	200 222 123 55 i
*> 100.0.0.0	100.1.1.1			0	100 i
*	200.1.1.1			0	200 2 100 i
*> 111.0.0.0	100.1.1.1			0	100 111 i
*	200.1.1.1			0	200 2 100 111 i
* 123.0.0.0	100.1.1.1			0	100 111 55 123 i
*>	200.1.1.1			0	200 222 123 i
*> 200.0.0.0/8	200.1.1.1			0	200 i
*> 222.0.0.0/8	200.1.1.1			0	200 222 i

2.3 Configurando conexões BGP com duas operadoras

- Detalhando as rotas recebidas via BGP
- O que significa essas 2 ”rotas alternativas” desaparecerem até da RIB?
- E o que isso implica na prática?

```
*  200.0.0.0/8    100.1.1.1                              0    100 2 200 i
*> 200.0.0.0/8    200.1.1.1                              0    200 i
*  222.0.0.0/8    100.1.1.1                              0    100 2 200 222 i
*> 222.0.0.0/8    200.1.1.1                              0    200 222 i
```

- Posteriormente detalharemos no DEBUG o processo que faz isso acontecer
- Por enquanto o importante à saber é que esse router passou a ser ”trânsito” dos AS's 200 e 222 para alcançar o AS 100,111 e todos os seus ”downstreams”
- No caso desse router ficar down, outro grupo ficará na mesma situação... vamos ver quem?

2.3 Configurando conexões BGP com duas operadoras

- Evitando se tornar AS de trânsito dos meus UPSTREAMS
- Fazer anúncios faz do meu router um potencial provedor de trânsito das redes que anuncio (dependendo da origem)

GX-R1>sh ip bgp neighbors 100.X.1.1 advertised-routes

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> X.0.0.0          0.0.0.0                  0         32768 i
*> 123.0.0.0        200.1.1.1                              0 200 222 123 i
*> 200.0.0.0/8      200.1.1.1                              0 200 ?
*> 222.0.0.0/8      200.1.1.1                              0 200 222 i
...
```

GX-R1>sh ip bgp neighbors 200.X.1.1 advertised-routes

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> X.0.0.0          0.0.0.0                  0         32768 i
*> 2.0.0.0          100.1.1.1                              0 100 2 i
*> 55.0.0.0         100.1.1.1                              0 100 111 55 i
*> 100.0.0.0        100.1.1.1                              0 100 i
*> 111.0.0.0        100.1.1.1                              0 100 111 i
...
```

2.3 Configurando conexões BGP com duas operadoras

- Implementando filtros de saída em meus upstreams
- Há várias maneiras e vamos iniciar pela mais simples

1) Criando uma prefix-list

```
GX-R1(config)#ip prefix-list MEUS-BLOCOS permit X.0.0.0/8
   !--- adiciona o bloco do AS do grupo na lista "MEUS-BLOCOS"
GX-R1(config)#router bgp X
GX-R1(config-router)#neighbor 200.X.1.1 prefix-list MEUS-BLOCOS out
   !---aplicar o mesmo filtro na saída do AS 200

GX-R1(config-router)#neighbor 100.X.1.1 prefix-list MEUS-BLOCOS out
   !--- aplicar o mesmo filtro na saída do AS 100
GX-R1(config-router)#end
GX-R1#clear ip bgp *
```

2.3 Configurando conexões BGP com duas operadoras

- **Verificando as rotas recebidas após o filtro de saída**
  - Agora o router anuncia para os UPSTREAMS apenas o necessário que é o prefixo /8 correspondente ao grupo

GX-R1>sh ip bgp neighbors 100.X.1.1 advertised-routes

BGP table version is 24, local router ID is 9.9.9.9
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> X.0.0.0	0.0.0.0	0		32768	i

GX-R1>**sh ip bgp neighbors 200.X.1.1 advertised-routes**

BGP table version is 24, local router ID is 9.9.9.9
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> X.0.0.0	0.0.0.0	0		32768	i

2.4
Utilizando um Looking Glass

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

2.4 Utilizando um Looking Glass

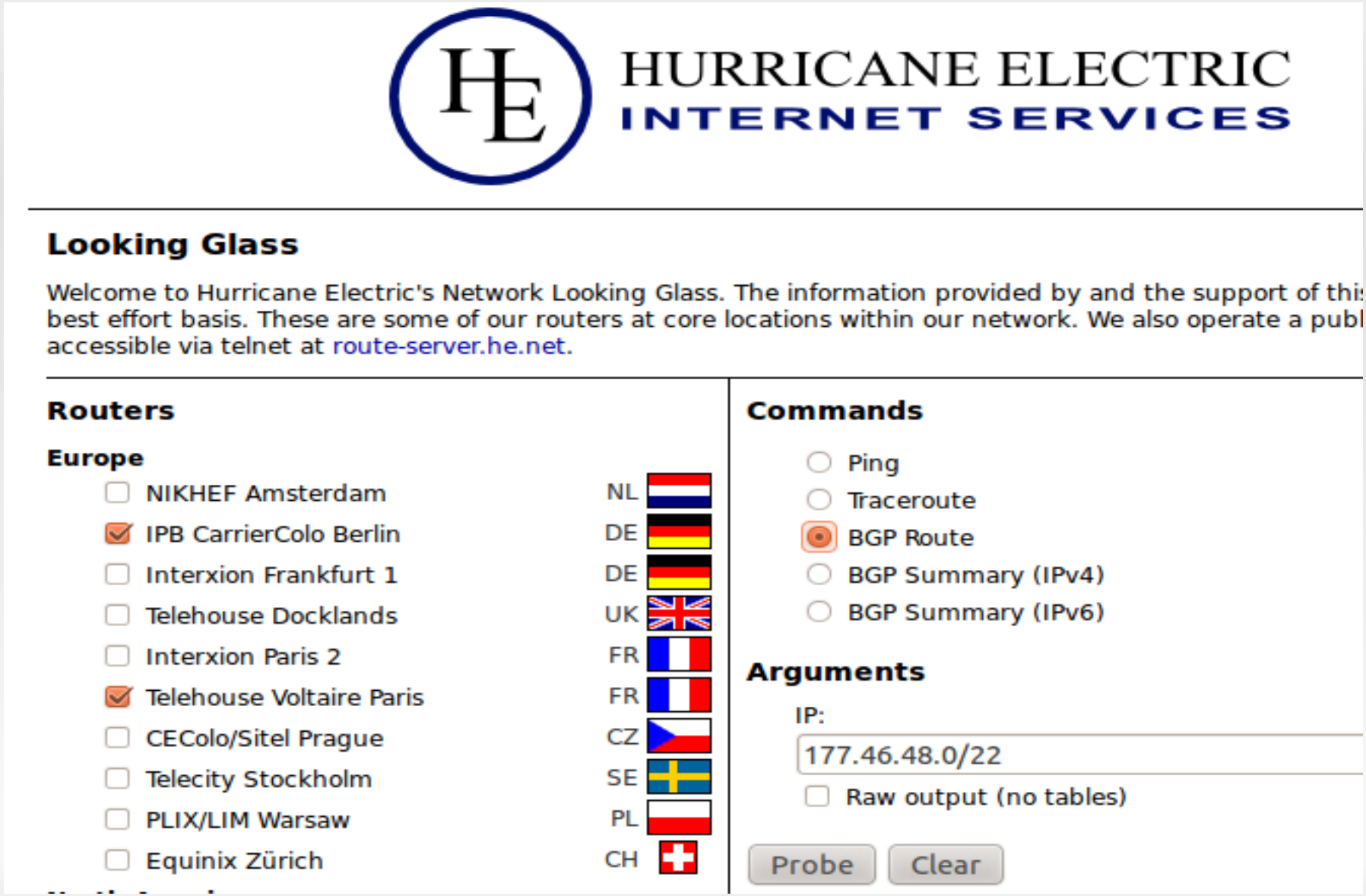
- Um looking glass nada mais é do que um router com acesso público em algum lugar da internet, em sua maioria, a interface é web para facilitar a vida do usuário, mas é o mesmo que entrar nesse router e executar certos comandos como:

show ip bgp x.x.x.x, ping x.x.x.x, traceroute x.x.x.x

Um dos mais famosos é o Hurricane Eletric...

http://lg.he.net

2.4 Utilizando um Looking Glass



2.4 Utilizando um Looking Glass

core1.ber1.he.net> show ip bgp routes detail 177.46.48.0/22

Matching Routes	5
Status Codes	**A** - Aggregate **B** - Best **b** - Not Install Best **C** - Confederation eBGP **D** - Damped **E** - eBGP **H** - History **I** - iBGP **L** - Local **M** - Multipath **m** - Not Installed Multipath **S** - Suppressed **F** - Filtered **s** - Stale

Status	Network	Next Hop	Metric	LocPr	Weight	Path	Origin
BI	177.46.48.0/22	195.69.144.196	150	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	195.66.224.153	230	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	80.81.192.194	220	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	198.32.176.70	1710	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	206.223.115.6	980	100	0	6762, 26615, 28135	IGP

Last Update 1d20h18m22s ago (1 path installed)

Entry cached for another 56 seconds.

core1.par2.he.net> show ip bgp routes detail 177.46.48.0/22

Matching Routes	5
Status Codes	**A** - Aggregate **B** - Best **b** - Not Install Best **C** - Confederation eBGP **D** - Damped **E** - eBGP **H** - History **I** - iBGP **L** - Local **M** - Multipath **m** - Not Installed Multipath **S** - Suppressed **F** - Filtered **s** - Stale

Status	Network	Next Hop	Metric	LocPr	Weight	Path	Origin
BI	177.46.48.0/22	195.66.224.153	85	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	195.69.144.196	110	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	80.81.192.194	100	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	198.32.176.70	1520	100	0	6762, 26615, 28135	IGP
I	177.46.48.0/22	206.223.115.6	790	100	0	6762, 26615, 28135	IGP

Last Update 1d20h18m28s ago (1 path installed)

Entry cached for another 57 seconds.

2.4 Utilizando um Looking Glass

- **Em nosso LAB o Looking glass é 123.123.123.1, vamos primeiramente verificar a conectividade com esse endereço:**

ALUNO@notebook:~$ traceroute 123.123.123.1 -n

traceroute to 123.123.123.1 (123.123.123.1), 30 hops max, 60 byte packets

1 1.0.1.254 5.292 ms 5.655 ms 5.640 ms

2 200.1.1.1 13.807 ms 14.198 ms 100.701 ms

3 200.200.1.1 111.035 ms 86.082 ms 86.477 ms

4 222.200.10.1 92.724 ms 92.302 ms 69.351 ms

5 222.222.1.1 127.492 ms 73.505 ms 69.707 ms

6 123.222.2.1 43.309 ms * *

***Clientes Windows utilizam "tracert 123.123.123.1"**

- O que podemos concluir com esse traceroute?
- Meus pacotes entram no AS 200, depois no AS 222 e chegam ao seu destino

2.4 Utilizando um Looking Glass

- Qual critério o meu router (primeiro salto) utilizou para considerar esse caminho como melhor?
- Vamos entender:

```
GX-R1>sh ip bgp 123.123.123.1
BGP routing table entry for 123.0.0.0/8, version 7
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  100 111 55 123
    100.X.1.1 from 100.X.1.1 (100.100.100.1)
      Origin IGP, localpref 100, valid, external
  200 222 123
    200.X.1.1 from 200.X.1.1 (200.200.200.1)
Origin IGP, localpref 100, valid, external, best
```

- Notem que em nenhuma das rotas há qualquer valor no tributo "weight" que é o primeiro critério, em segundo é verificado o "localpref" que nos dois casos está em 100 (empate), o algoritmo de seleção de rotas segue então para o próximo "critério de desempate" (menor AS Path) onde a rota #2 foi mais feliz

2.4 Utilizando um Looking Glass

#Conectar-se ao looking Glass

ALUNO@notebook:~$ telnet 123.123.123.1

AS-123-R1>traceroute X.128.0.2
Type escape sequence to abort.
Tracing the route to 1.0.1.1
1 123.222.2.2 8 msec 8 msec 12 msec
2 222.222.1.2 [AS 222] 12 msec 40 msec 12 msec
3 222.200.10.2 [AS 222] 24 msec 28 msec 12 msec
4 200.200.1.2 [AS 200] 52 msec 24 msec 56 msec
5 200.X.1.2 [AS 200] 56 msec 24 msec 28 msec
6 X.128.0.2 [AS X] 64 msec 64 msec 28 msec

- **Os pacotes seguiram o seguinte caminho: AS 222 depois AS 200 e finalmente AS X**

2.4 Utilizando um Looking Glass

- **Verifique agora que o mesmo critério (menor AS Path) fez o router decidir pelo "mesmo caminho" para alcançar o notebook do aluno**

AS-123-R1>sh ip bgp X.128.0.2
BGP routing table entry for X.0.0.0/8, version 45
Paths: (2 available, best #2, table Default-IP-Routing-Table)
 Advertised to non peer-group peers:
 123.55.2.2
 55 111 100 X
  123.55.2.2 from 123.55.2.2 (55.55.55.2)
   Origin IGP, localpref 100, valid, external
 222 200 X
  123.222.2.2 from 123.222.2.2 (222.222.222.2)
   Origin IGP, localpref 100, valid, external, best

2.5
Fail-Over automático

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

2.5 Fail-Over automático

- Agora execute novamente um tracerotue a partir do seu laptop

```
ALUNO@notebook:~$ traceroute 123.123.123.1 -n
traceroute to 123.123.123.1 (123.123.123.1), 30 hops max, 60 byte packets
 1  X.128.0.1  3.118 ms  3.492 ms  3.482 ms
 2  100.X.1.1  11.808 ms  11.799 ms  17.716 ms
 3  100.100.1.1  38.856 ms  42.330 ms  21.847 ms
 4  111.100.10.1  38.422 ms  108.815 ms  108.392 ms
 5  111.111.1.1  133.799 ms  69.868 ms  133.367 ms
 6  55.111.2.1  110.819 ms  107.647 ms  65.948 ms
 7  55.55.1.1  138.345 ms  129.942 ms  129.917 ms
 8  123.55.2.1  115.805 ms * *
ALUNO@notebook:~$
```

- Com o AS 200 down os pacotes entram no AS 100, depois no AS 111, e só depois de passar no AS 55 alcançam o destino (looking glass)

2.5 Fail-Over automático

- Utilizando a rota alternativa

- Vejamos o que acontece quando um dos links fica fora colocando em shutdown a sessão BGP com o AS 200

```
GX-R1>enable
GX-R1#conf t
GX-R1(config)#router bgp x
GX-R1(config-router)#neighbor 200.X.1.1 shutdown
GX-R1(config-router)#
5d06h: %BGP-5-ADJCHANGE: neighbor 200.X.1.1 Down Admin. shutdown
```

- aguardem alguns instantes...

2.5 Fail-Over automático

- Agora note que a rota alternativa (mais distante) passou a ser utilizada e o "melhor caminho" que havia não existe mais, restando apenas o "pior caminho" que dadas as circunstâncias, passou a ser um "ótimo" caminho

```
GX-R1>sh ip bgp 123.123.123.1
BGP routing table entry for 123.0.0.0/8, version 16
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  100 111 55 123
    100.X.1.1 from 100.X.1.1 (100.100.100.[1-4])
      Origin IGP, localpref 100, valid, external, best
```

2.5 Fail-Over automático

- Vamos verificar no ponto de vista do LG

```
ALUNO@notebook:~$ telnet 123.123.123.1
Trying 123.123.123.1...
Connected to 123.123.123.1.
Escape character is '^]'.


User Access Verification

Password:
AS-123-R1>traceroute X.128.0.2

Type escape sequence to abort.
Tracing the route to 1.0.1.1

  1 123.55.2.2 8 msec 8 msec 4 msec
  2 55.55.1.2 [AS 55] 12 msec 24 msec 16 msec
  3 55.111.2.2 [AS 55] 32 msec 28 msec 40 msec
  4 111.111.1.2 [AS 111] 52 msec 36 msec 64 msec
  5 111.100.10.2 [AS 111] 56 msec 48 msec 52 msec
  6 100.100.1.2 [AS 100] 84 msec 60 msec 64 msec
  7 100.X.1.2 [AS 100] 84 msec 88 msec 60 msec
  8 X.128.0.2 [AS 1] 92 msec 68 msec 96 msec
```

2.5 Fail-Over automático

- Verifiquem que o looking glass escolhe o caminho mais longo para chegar ao notebook do aluno

  *agora é a única rota disponível

AS-123-R1>sh ip bgp X.128.0.2
BGP routing table entry for 1.0.0.0/8, version 49
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  123.222.2.2
  55 111 100 **X**
    123.55.2.2 from 123.55.2.2 (55.55.55.2)
      Origin IGP, localpref 100, valid, external, best

- E essa mensagem "advertised to... 123.222.2.2" ?

  Nos slides anteriores era "advertised to 123.55.2.2"

- Esses detalhes serão vistos nos capítulos posteriores
- Mas deixa claro que a(s) rede(s) do(s) aluno(s) estão sendo anunciadas para o as 222, e consequentemente para o AS 200
- Vamos comprovar...

2.5 Fail-Over automático

- Partindo do laptop vamos tentar alcançar o AS 200...

```
ALUNO@notebook:~$ traceroute 200.200.200.10 -n
traceroute to 200.200.200.10 (200.200.200.10), 30 hops max, 60 byte packets
 1  X.128.0.1  6.128 ms  6.510 ms  6.501 ms
 2  100.X.1.1  19.342 ms  19.735 ms  68.139 ms
 3  100.100.1.1  100.368 ms  68.527 ms  99.945 ms
 4  111.100.10.1  68.506 ms  79.031 ms  100.321 ms
 5  111.111.1.1  78.604 ms  68.467 ms  68.456 ms
 6  55.111.2.1  45.810 ms  39.272 ms  127.491 ms
 7  55.55.1.1  136.322 ms  123.845 ms  114.650 ms
 8  123.55.2.1  113.618 ms  123.932 ms  112.493 ms
 9  123.222.2.2  161.647 ms  101.636 ms  111.676 ms
10  222.222.1.2  161.596 ms  176.097 ms  183.647 ms
11  222.200.10.2  154.398 ms * *
```

- Agora alcançamos o AS 200 pelo caminho mais longo possível, AS 100, AS 111, AS 55, AS 123, AS 222 e finalmente AS 200
- Vamos analizar...

2.5 Fail-Over automático

- Verifiquem as entradas para a rede em que faz parte o IP do AS-200-R10

```
GX-R1>sh ip bgp 200.200.200.10
BGP routing table entry for 200.0.0.0/8, version 20
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  100 111 55 123 222 200
    100.X.1.1 from 100.X.1.1 (100.100.100.1)
      Origin incomplete, localpref 100, valid, external, best
```

- Observe o AS Path, precisamos dar a "volta ao mundo" para alcançar o AS 200
- Discutiremos soluções alternativas nos próximos capítulos

2.5 Fail-Over automático

- A mesma coisa pode ser observada a partir do R10 do AS 200

```
ALUNO@notebook:~$ telnet 200.200.200.10
Trying 200.200.200.10
Connected to 200.200.200.10
Escape character is '^]'.
Connected to Dynamips VM "AS-200-R10" (ID 3, type c3600) - Console port

AS-200-R10>sh ip bgp X.0.0.0
BGP routing table entry for 1.0.0.0/8, version 77
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to peer-groups:
     IBGP
 222 123 55 111 100 X
   222.200.10.1 from 222.200.10.1 (222.222.222.1)
      Origin IGP, localpref 100, valid, external, best
```

- O novo caminho fica bem mais longo, mas não fica fora

2.5 Fail-Over automático

Conclusões:

- O protocolo BGP trata praticamente sozinho os problemas de queda nos links
- Porém não tem como entender certas variáveis que fazem o link A ser mais viável que o link B, ou mesmo quando precisamos somar 2 links
- Em outras palavras, se tenho contratados 100Mbs de trânsito com o AS 100 e apenas 50 com o AS 200, seria inviável ter o link do AS 200 como principal, e nesse caso precisamos "interferir" na decisão do protocolo BGP manipulando certos atributos.

2.6
ESCOLHENDO O MELHOR LINK

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

2.6 Escolhendo o melhor link

- Considere quase toda a minha demanda de trânsito vindo sempre do AS 123 as seguintes situações:

- O link entre o AS 222 e 200 está passando por dificuldades

- A negociação com o AS 100 me garantiu um preço que me permitiu dobrar a banda contratada.

- Algum outro motivo que me obrigue a utilizar o AS 100 como link principal e o as 200 como backup
- Como vimos anteriormente, o BGP escolhe por padrão o as 200, vamos modificar essa escolha...

2.6 Escolhendo o melhor link

- Alterando o atributo Weight:

```
Gx-R1#configure terminal
Gx-R1(config)#router bgp X
Gx-R1(config-router)#neighbor 100.X.1.1 weight 10
Gx-R1(config-router)#end
!--- agora vamos efetuar um clear na sessão BGP
Gx-R1#clear ip bgp 100.X.1.1 soft
```

- Vamos verificar como ficaram as rotas BGP...

2.6 Escolhendo o melhor link

- Analizando a rota para rede do looking glass após a mudança no atributo

GX-R1#sh ip bgp 123.123.123.1
BGP routing table entry for 123.0.0.0/8, version 35
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer

200 222 123
  200.X.1.1 from 200.1.1.1 (200.200.200.1)
    Origin IGP, localpref 100, valid, external

100 111 55 123
  100.X.1.1 from 100.X.1.1 (100.100.100.1)
    Origin IGP, localpref 100, **weight 10**, valid, external, **best**

- Acabamos de forçar o router a escolher a rota #2, já que o primeiro critério é o atributo weight, a rota 2# foi escolhida como melhor mesmo com um as path MAIOR

2.6 Escolhendo o melhor link

- Testando...
- Executem um traceroute do seu laptop para 123.123.123

  …
  ...
- Caso a mudança no weight tenha sido bem sucedida os pacotes seguiram a rota pelo AS 100 ao invés do AS 200
- Porém isso resolve apenas "metade" do meu problema conforme veremos no slide seguinte

2.6 Escolhendo o melhor link

- Testando...
- Executem um traceroute do seu laptop para 123.123.123.1

  …
  ...
- Caso a mudança no weight tenha sido bem sucedida os pacotes seguiram a rota pelo AS 100 ao invés do AS 200
- Porém isso resolve apenas "metade" do meu problema conforme veremos no slide seguinte

2.6 Escolhendo o melhor link

- Agora verificar o ponto de vista do looking glass testando a partir dele:

ALUNO@notebook:~ telnet 123.123.123.1

!--- Efetuem um traceroute para o seu laptop

AS-123-R1>traceroute para X.128.0.2

Como podemos verificar, o AS 123 continua escolhendo a mesma rota, uma vez que a mudança no atributo weight, teve influência apenas no meu UPLOAD, sendo assim, meu download continua chegando pelo AS 200 (menor patch no ponto de vista do AS 123)

- Não tenho como pedir que o AS 123 altere os seus atributos, preciso convence-los de outra forma a mudar de rota

2.6 Escolhendo o melhor link

- Fazendo anuncios mais específicos para o AS 100
- Como mencionado anteriormente, os atributos BGP só são analizados quando há um empate de prefixo, porém quando há um prefixo MAIS ESPECÍFICO os atrubutos são ignorados
- Quais blocos anunciar para cada AS?
- Dividiremos em 2 blocos /9 para o as 100 e o /8 para o AS 200

2.6 Escolhendo o melhor link

- Configurando os anúncios mais específicos

```
GX-R1#conf terminal
GX-R1(config)#ip prefix-list ANUNCIAR-AS-100 permit X.0.0.0/9
GX-R1(config)#ip prefix-list ANUNCIAR-AS-100 permit X.128.0.0/9
!--- adiciona 2 prefixos /9 dentro da lista ”ANUNCIAR-AS-100”

GX-R1(config)#ip prefix-list ANUNCIAR-AS-200 seq 10 permit x.0.0.0/8
!--- adiciona o classe A inteiro dentro da lista ”ANUNCIAR-AS-100”
GX-R1(config)#router bgp X
!--- adiciona 2 prefixos /9 dentro da lista ”ANUNCIAR-AS-100”
GX-R1(config-router)#network X.0.0.0 mask 255.128.0.0
GX-R1(config-router)#network X.128.0.0 mask 255.128.0.0
!--- além do anuncio local do classe A, inserimos mais 2 prefixos /9 a serem
!--- a serem anunciados pelo router

GX-R1(config-router)#neighbor 200.X.1.1 prefix-list ANUNCIAR-AS-200 out
!--- declara que esse anuncios de saida (out) para esse neighbor serao filtrados

GX-R1(config-router)#neighbor 100.X.1.1 prefix-list ANUNCIAR-AS-100 out
!--- declara que esse anuncios de saida (out) para esse neighbor serao filtrados
GX-R1(config-router)#end

GX-R1#clear ip bgp * soft
```

2.6 Escolhendo o melhor link

- Verificando anúncios para o AS 200:

```
Gx-R1#sh ip bgp neighbors 200.x.1.1 advertised-routes
BGP table version is 13, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> x.0.0.0          0.0.0.0                  0         32768 i
Gx-R1#
```

- Agora para o AS 100:

```
Gx-R1#sh ip bgp neighbors 100.x.1.1 advertised-routes

 Network          Next Hop            Metric LocPrf Weight Path
*> x.128.0.0/9     0.0.0.0                  0         32768 i
```

- Estamos anunciando apenas um dos blocos /9 para o AS 100, por que?

2.6 Escolhendo o melhor link

- Apenas um prefixo pertencente ao meu classe A está na FIB

```
Gx-R1#sh ip route x.0.0.0 longer-prefixes
...

Gateway of last resort is not set

     x.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       x.x.x.x/32 is directly connected, Loopback0
C       x.128.0.0/9 is directly connected, FastEthernet0/0
```

- O comando ”longer-prefixes” mostraria rotas para qualquer sub-rede dentro do bloco /8 pesquisado, porém temos o primeiro /9 diretamente conectado à f0/0 e um /32 na Loopback0

2.6 Escolhendo o melhor link

- Precisamos primeiramente garantir que as redes que quero anunciar estejam em minha FIB

```
Gx-R1#conf t
Gx-R1(config)#ip route x.0.0.0 255.128.0.0 null0
!--- faremos apenas para o primeiro /9 já que o segundo está OK
Gx-R1(config)#end
Gx-R1#
Gx-R1#sh ip route x.0.0.0 longer-prefixes
...
Gateway of last resort is not set

     x.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       x.x.x.x/32 is directly connected, Loopback0
S       x.0.0.0/9 is directly connected, Null0
C       x.128.0.0/9 is directly connected, FastEthernet0/0
```

- Agora que tenho os 2 blocos /9 em minha FIB vamos verificar novamente os anúncios para o AS 100...

2.6 Escolhendo o melhor link

- Agora que tenho os 2 blocos /9 em minha FIB vamos verificar novamente os anúncios para o AS 100...

```
Gx-R1#sh ip bgp neighbors 100.x.1.1 advertised-routes
BGP table version is 11, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> x.0.0.0/9        0.0.0.0                  0         32768 i
*> x.128.0.0/9      0.0.0.0                  0         32768 i
Gx-R1#
```

- Agora estamos fazendo corretamente o anúncio dos dois blocos /9 para o AS 100

2.6 Escolhendo o melhor link

- Vamos novamente entrar no looking Glass e verificar as rotas para classe A do grupo:

```
AS-123-R1>sh ip bgp x.0.0.0/8 longer-prefixes
BGP table version is 30, local router ID is 123.123.123.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> x.0.0.0/9        123.55.2.2                             0 55 111 100 x i
*> x.0.0.0          123.222.2.2                            0 222 200 x i
*> x.128.0.0/9      123.55.2.2                             0 55 111 100 x i
AS-123-R1>
```

- Dessa forma, a rota em vermelho nunca é usada, pois existem dois blocos mais específicos, fazendo com que o LG sempre alcance os AS's dos grupos através do AS 100

2.6 Escolhendo o melhor link

- -Verificando caminho percorrido com o comando traceroute:

```
AS-123-R1>traceroute x.128.0.2

Type escape sequence to abort.
Tracing the route to x.128.0.2

  1 123.55.2.2 20 msec 24 msec 12 msec
  2 55.55.1.2 [AS 55] 32 msec 32 msec 44 msec
  3 55.111.2.2 [AS 55] 20 msec 36 msec 32 msec
  4 111.111.1.2 [AS 111] 40 msec 56 msec 36 msec
  5 111.100.10.2 [AS 111] 88 msec 28 msec 56 msec
  6 100.100.1.2 [AS 100] 76 msec 52 msec 76 msec
  7 100.2.1.2 [AS 100] 64 msec 56 msec 60 msec
  8 x.128.0.2 [AS x] 60 msec 60 msec 80 msec
AS-123-R1>
```

- No início forçamos o upload pelo AS 100, e agora também o download pelo AS 100

2.7
BALANCEANDO OS LINKS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

2.7-Balanceando os links

- Vamos agora considerar uma situação de necessidade de utilizar os dois links de maneira balanceada
- Dividiremos cada metade do /8 para cada AS
- Anunciaremos o primeoro /9 para o AS 100 e o segundo /9 para o AS 200 fazendo com que o nosso tráfego inbound seja divido em 50% para cada link
- A lógica é a mesma, o que faremos diferente do lab anterior é um detalhe para o fail-over

2.7-Balanceando os links

- Já temos as prefix-lists declaradas para cada neighbor
- Vamos apenas mudar o seu conteudo:

```
Gx-R1#conf terminal

Gx-R1(config)#no ip prefix-list ANUNCIAR-AS-100
!--Apagar primeiro a antiga prefix-list para evitar inconsistências

Gx-R1(config)#ip prefix-list ANUNCIAR-AS-100 permit x.0.0.0/9
!-- primeiro  /9 dentro do classe A
Gx-R1(config)#ip prefix-list ANUNCIAR-AS-100 permit x.0.0.0/8
!-- Classe A inteiro
```

2.7-Balanceando os links

- A mesma lógica aplicada para o AS 200, porém permitiremos apenas o anúncio do último /9 do classe A

```
Gx-R1(config)#no ip prefix-list ANUNCIAR-AS-200
!--- apagar primeiro a antiga prefix-list para evitar inconsistências

Gx-R1(config)#ip prefix-list ANUNCIAR-AS-200 permit x.128.0.0/9
!--- segundo /9 dento do classe A
Gx-R1(config)#ip prefix-list ANUNCIAR-AS-200 permit x.0.0.0/8
!--- classe A inteiro
```

- Como já adicionamos a rota estática que faltava, já temos os dois /9 na FIB e não precisamos mais adicionar rotas esticas

2.7-Balanceando os links

- Vamos entrar no looking glass e ver o resultado:

AS-123-R1>sh ip bgp X.0.0.0/8 longer-prefixes

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> X.0.0.0/9        123.55.2.2                             0 55 111 100 X i
*   X.0.0.0         123.55.2.2                             0 55 111 100 X i
*>                  123.222.2.2                            0 222 200 X i
*> X.128.0.0/9      123.55.2.2                             0 222 200 X i
AS-123-R1>
```

- Agora testar o caminho para os 2 blocos /9

2.7-Balanceando os links

- Primeiro /9 (Loopback do router)

AS-123-R1>traceroute X.X.X.X

Type escape sequence to abort.
Tracing the route to X.X.X.X

1 123.55.2.2 8 msec 16 msec 16 msec
2 55.55.1.2 [AS 55] 44 msec 12 msec 16 msec
3 55.111.2.2 [AS 55] 24 msec 32 msec 32 msec
4 111.111.1.2 [AS 111] 48 msec 28 msec 48 msec
5 111.100.10.2 [AS 111] 48 msec 48 msec 56 msec
6 100.100.1.2 [AS 100] 60 msec 88 msec 112 msec
AS-123-R1>

- Caminho 55,111,100

2.7-Balanceando os links

- Segundo /9 (ip do laptop)

```
AS-123-R1#traceroute X.128.0.2

Type escape sequence to abort.
Tracing the route to X.128.0.1

  1 123.222.2.2 24 msec 4 msec 32 msec
  2 222.222.1.2 [AS 222] 40 msec 8 msec 16 msec
  3 222.200.10.2 [AS 222] 32 msec 24 msec 16 msec
  4 200.200.1.2 [AS 200] 56 msec 4 msec 28 msec
  5 200.2.1.2 [AS 200] 92 msec 20 msec 40 msec
  8 2.128.0.2 [AS 2] 64 msec 47 msec 33 msec
AS-123-R1#
```

- Caminho 222,200

2.7-Balanceando os links

- Vamos todos derrubar a sessão BGP com o as 100 e verificar o funcionamento do fail-over a partir do looking glass
- Em seguida derrubar a sessão BGP com o AS 200 e fazer a mesma verificação
- OBS: Para um balanceamento simétrico mudaríamos os anúncios, sendo o primeiro /9 + /8 para o AS100 e o segundo /9 + /8 para o AS200
- Nunca esquecer de que o balanceamento baseado em anúncios específicos depende da distribuição dos IPs dentro do meu AS

3.1
PEERING x TRÂNSITO

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.1 Peering x Trânsito

- Em primeiro lugar devem ser separados os conceitos de peering e trânsito
- Quando dois AS's tem uma quantidade de demanda comum, acaba saindo mais barato o peering
- Exemplo, eu sou AS A, e tenho uma demanda de 10Mbs de download de um certo conteúdo hospedado no AS B
- Para resolver isso, contrato um link de 10Mbs da operadora B e pago R$ 100,00 por cada mega, num total de R$ 1.000,00

3.1 Peering x Trânsito

- Porem, a operadora B também tem uma demada de 10Mbs de conteúdo hospedado no meu AS A.
- Isso dá para operadora B uma conta semelhante de R$ 1.000,00 mensais
- No final do mês, ficaríamos empatados no custo, e o melhor a se fazer é estabelecer um link direto de 10Mbs sem que um cobre do outro
- Há outras vantagens técnicas como menor caminho BGP e menor latência

3.1 Peering x Trânsito

- LAB: Estabelecer peering entre grupos com os seguintes IPs

AS	IP		AS	IP
1	1.2.1.1/30	x	2	1.2.1.2/30
3	3.4.1.1/30	x	4	3.4.1.2/30
5	5.6.1.1/30	x	6	5.6.1.2/30
7	7.8.1.1/30	x	8	7.8.1.2/30
9	9.10.1.1/30	x	10	9.10.1.2/30
11	11.12.1.1/30	x	12	11.12.1.2/30
13	13.14.1.1/30	x	14	13.14.1.2/30
15	15.16.1.1/30	x	16	15.16.1.2/30
17	17.18.1.1/30	x	18	17.18.1.2/30
19	19.20.1.1/30	x	20	19.20.1.2/30
21	21.22.1.1/30	x	22	21.22.1.2/30
23	23.24.1.1/30	x	24	23.24.1.2/30
-	-	-	-	-

3.1 Peering x Trânsito

- Realizando as configurações IP (AS IMPAR)

  GXR1#conf t

  GX-R1(config)#interface serial 1/2

  GX-R1(config-if)#ip address X.Y.1.1 255.255.255.252

  GX-R1(config-if)#no shutdown

  GX-R1(config-if)#end

  GXR1#ping X.Y.1.2

- Onde:

  X= AS local

  Y= AS remoto

3.1 Peering x Trânsito

- Realizando as configurações IP (AS PAR)

  GXR1#conf t

  GX-R1(config)#interface serial 1/2

  GX-R1(config-if)#ip address Y.X.1.2 255.255.255.252

  GX-R1(config-if)#no shutdown

  GX-R1(config-if)#end

  GXR1#ping Y.X.1.1

- Onde:

  X= AS local

  Y= AS remoto

3.1 Peering x Trânsito

- Configurando BGP (AS ímpar)

  - Onde:
    X= AS local
    Y= AS remoto

```
GX-R1#conf t
GX-R1(config)#
GX-R1(config)#router bgp X
GX-R1(config-router)#neighbor X.Y.1.2 remote-as Y
GX-R1(config-router)#neighbor X.Y.1.2  weight 20
!--- priorizar todas as rotas recebidas pelo peering com weight 20
GX-R1(config-router)#neighbor X.Y.1.2 filter-list 1 out
!--- filtrar anúncios de saída para esse peering na filter-list 1
GX-R1(config-router)#exit
GX-R1(config)#ip as-path access-list 1 permit ^$
!--- adiciona na filter-list 1 (access-list) permissão para qualquer
!--- rota de origem LOCAL
GX-R1(config)#end
GX-R1#clear ip bgp * soft
```

3.1 Peering x Trânsito

- Configurando BGP (AS par)

  - Onde:
    X= AS local
    Y= AS remoto

```
GX-R1#conf t
GX-R1(config)#
GX-R1(config)#router bgp X
GX-R1(config-router)#neighbor Y.X.1.1 remote-as Y
GX-R1(config-router)#neighbor Y.X.1.1  weight 20
!--- priorizar todas as rotas recebidas pelo peering com weight 20
GX-R1(config-router)#neighbor Y.X.1.1 filter-list 1 out
!--- filtrar anúncios de saída para esse peering na filter-list 1
GX-R1(config-router)#exit
GX-R1(config)#ip as-path access-list 1 permit ^$
!--- adiciona na filter-list 1 (access-list) permissão para qualquer
!--- rota de origem LOCAL
GX-R1(config)#end
GX-R1#clear ip bgp * soft
```

3.1 Peering x Trânsito

- Verificando os anúncios para o novo peering

GX-R1#sh ip bgp neighbors X.X.X.X advertised-routes

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.0.0.0/9        0.0.0.0                  0             32768 i
*> 1.0.0.0          0.0.0.0                  0             32768 i
*> 1.128.0.0/9      0.0.0.0                  0             32768 i
```

!-- x.x.x.x é o endereço remoto do seu peering

- O Next Hop 0.0.0.0 e o weight 32768 indicam que as redes foram originadas localmente ou seja, no comando:
  router(config-router) network x.x.x.x mask x.x.x.x

3.1 Peering x Trânsito

- Verificando as redes recebidas do meu novo peering (utilizando o exemplo para o peering entre AS 1 e 2)

```
G1-R1#sh ip bgp regexp ^2$
!--- essa consulta retorna apenas rotas que contenham o AS 2 no path
!--- e nada mais
BGP table version is 29, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/9        1.2.1.2                  0             20 2 i
*> 2.0.0.0          1.2.1.2                  0             20 2 i
*> 2.128.0.0/9      1.2.1.2                  0             20 2 i
```

- Onde:
  1= AS local
  2= AS remoto

3.1 Peering x Trânsito

- Testando um traceroute para **Y**.0.0.1 ...
- A partir de agora está havendo uma troca de tráfego entre os 2 ASs
- O que faz dessa configuração um peering e não um trânsito?

  *Ao cairem os links do AS200 e AS100 apenas as redes do peering continuariam funcionando, e isso é o que descaracteriza um trânsito
- Vamos fazer esse teste desativando as interfaces dos links AS100 e AS200...

3.2
ENTENDENDO O FUNCIONANDO DE UM PIX DO PTT-METRO

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.2 Entendendo o funcionamento de um PIX do PTTMetro

- Para quem ainda não assistiu:

  http://youtu.be/imL3Cc8mC2Y

3.2 Entendendo o funcionamento de um PIX do PTTMetro

- Entendendo a necessidade de peerings entre centenas de ASs do Brasil, o NIC.br organizou diversos pontos de troca

- Seu funcionamento consiste basicamente em 1 switch, 2 route-servers e um router Looking Glass (outros maiores como SP possuem dezenas de switchs em vários PIXes)

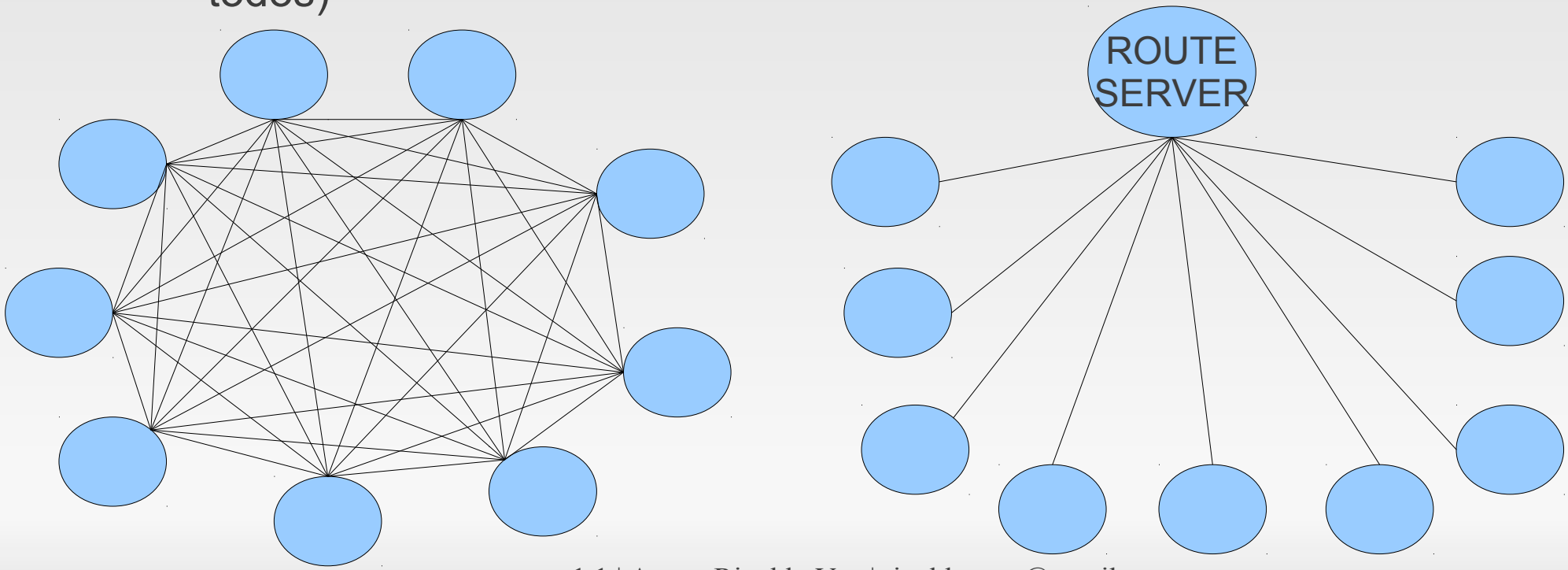
- Todos os participantes compartilham um mesmo bloco /24, cada participante tem um IP, e todos pingam entre si

- Estabelecem conexões BGP com os route-servers e o looking glass

3.2 Entendendo o funcionamento de um PIX do PTTMetro

- Os Route Servers cumprem um papel fundamental dispensando a necessidade de que todos os participantes façam peering individualmente

- Imaginem centenas de participantes estabelecendo peering (todos com todos)

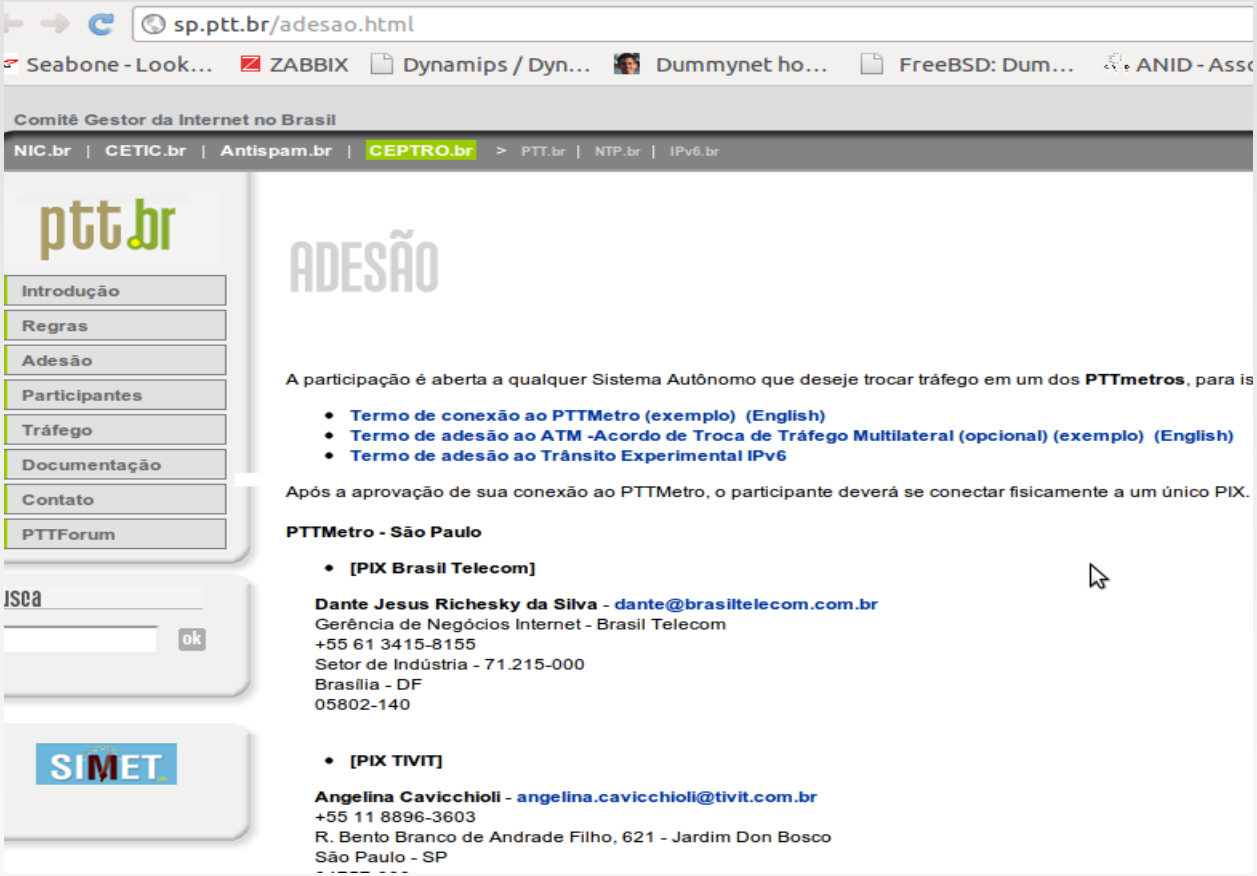


3.3
FORMULÁRIOS DE ADESÃO

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.3 - Formulários de adesão

- Para se conectar do PTT-SP por exemplo, acesse o endereço http://sp.ptt.br/adesao.html



3.3 - Formulários de adesão

- Para facilitar há um formulário modelo já preenchido:

```
...
                    Termo de conexão ao PTTMetro
...
Localidade ........: (Coloque aqui a localidade do PTT que deseja se conectar) Listagem: http://ptt.br/adesao.php
ASN ...............: 22548
Entidade Nome .....: Registro.br
URL ...............: http://registro.br/
Endereço ..........: Av. das Nações Unidas 11548, 7° andar 04578-000 - São Paulo - SP
NOC email .........: noc@registro.br
INOC-DBA ext ......: 22548*NOC
Telefone ..........: 11 5509-3510
Adm-c nome ........: Frederico A C Neves
email .............: fneves@registro.br
Telefone ..........: 11 5509-3511
Peering-c nome ....: Oripide Ocilento Filho
email .............: oripide@nic.br
Telefone ..........: 11 5509-3520
Comercial Trânsito :
email .............:
Telefone ..........:
URL Política
    Peering .......: none - Open Policy
BGP Feed para o
  Looking-glass ...: Sim [x]    Nao [ ]
Vende trânsitov4 ..: Sim [ ]    Não [x]
Vende trânsitov6 ..: Sim [ ]    Não [x]
Conexão
Bandwidth .........:
Media Ethernet ....: 10BaseTX  [ ]  100BaseTX [ ]  10000BaseTX [ ]
                     1 GigE-SX [X]  1 GigE-LX [ ]   10 GigE-SR [ ]  10 GigE-LR [ ]
PIX ...............: Registro.br
IPv6 ..............: Sim [x]    Nao [ ]
...
```

3.4
DIFERENÇAS ENTRE ACORDO BILATERAL E MULTILATERAL

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.4 - Diferenças entre acordos bilaterais e multilaterais

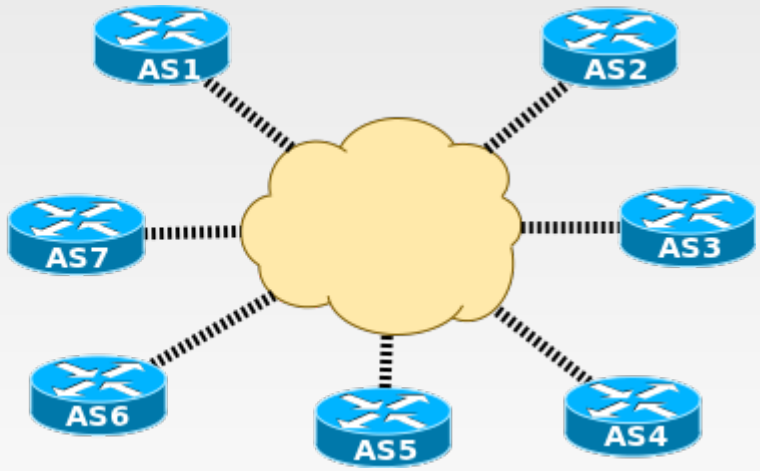
- Estar no PTT por si só não significa trocar tráfego;
- Existem participantes que optaram em não trocar tráfego multilateral (todos com todos)
- E esse é um direito do participante...

PTT - São Paulo								
ASN	NOME	ATM		LG		TRÂNSITO		IPV6
		V4	V6	V4	V6	V4	V6	
42	WoodyNet			✓				
1251	ANSP	✓	✓	✓	✓			✓
1916	RNP	✓	✓					✓
2688	AT&T	✓		✓		✓		
3549	Global Crossing			✓	✓	✓	✓	✓
3856	PCH			✓				
6447	RouteViews	✓	✓	✓	✓			✓
6762	Seabone-Lanautilus					✓	✓	
7048	Linha Livre	✓		✓				
7162	UOL Diveo		✓		✓			✓
7738	Oi					✓		
8055	Value4Net	✓	✓	✓	✓	✓	✓	✓
8167	Brasil Telecom					✓		
10362	UBIK	✓		✓				
10429	Telefonica			✓		✓		✓
10733	Matrix	✓	✓	✓	✓	✓	✓	✓

3.4 - Diferenças entre acordos bilaterais e multilaterais

- Acordo de troca Multilateral (ATM)
- Para trocar tráfego com todos os participantes, além do termo de conexão ao PTTmetro, é preciso preencher o termo de adesão ao ATM (Acordo de Troca Multilateral)
- No acordo multilateral o participante recebe uma vlan+ip e troca tráfego com todos

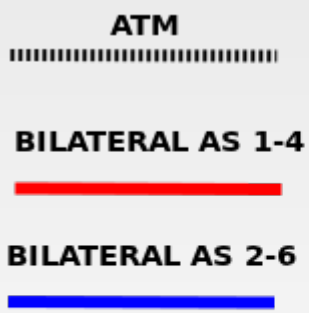
```
...
   Termo de adesão ao Acordo de Troca de Tráfego Multilateral (ATM)
Version .........: 2008071000
Este termo de adesão trata das informações e compromissos da entidade
solicitante para a participação no acordo de troca de tráfego
multilateral no PTTMetro.
Localidade ......: sp.ptt.br
ASN .............: 22548
Entidade Nome ...: Registro.br
ATM v4 ..........: Sim [x]    Não [ ]
ATM v6 ..........: Sim [x]    Não [ ]
...
```

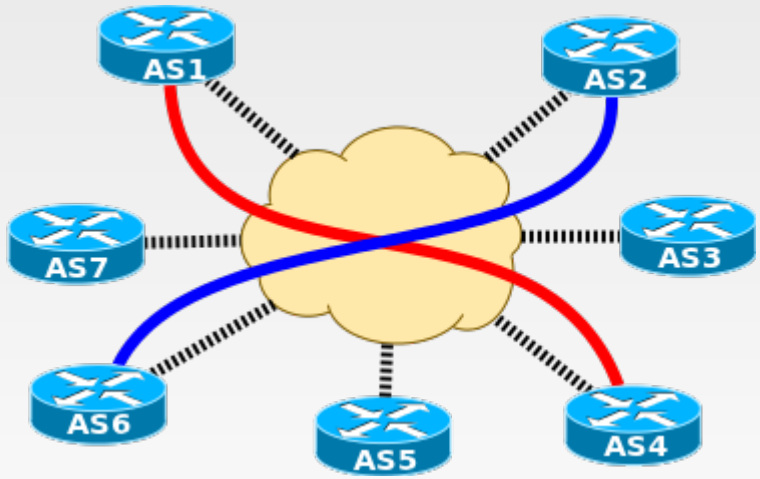


- Após preenchidos ambos devem ser encaminhados para join@ptt.br

3.4 - Diferenças entre acordos bilaterais e multilaterais

- Acordo de troca Bilateral
- No acordo bilateral, 2 participantes manifestam interesse e recebem uma vlan comum, que fica extendida às duas extremidades e, por se tratar de um serviço em layer 2, os IPs ficam à critério desses 2 participantes
- Em outras palavras, o PTTmetro providencia toda estrutura de um ”peering” para ambos





- Os AS's que se conectam ao PTT mas não participam do ATM, em geral vendem trânsivo através de acordos bilaterais

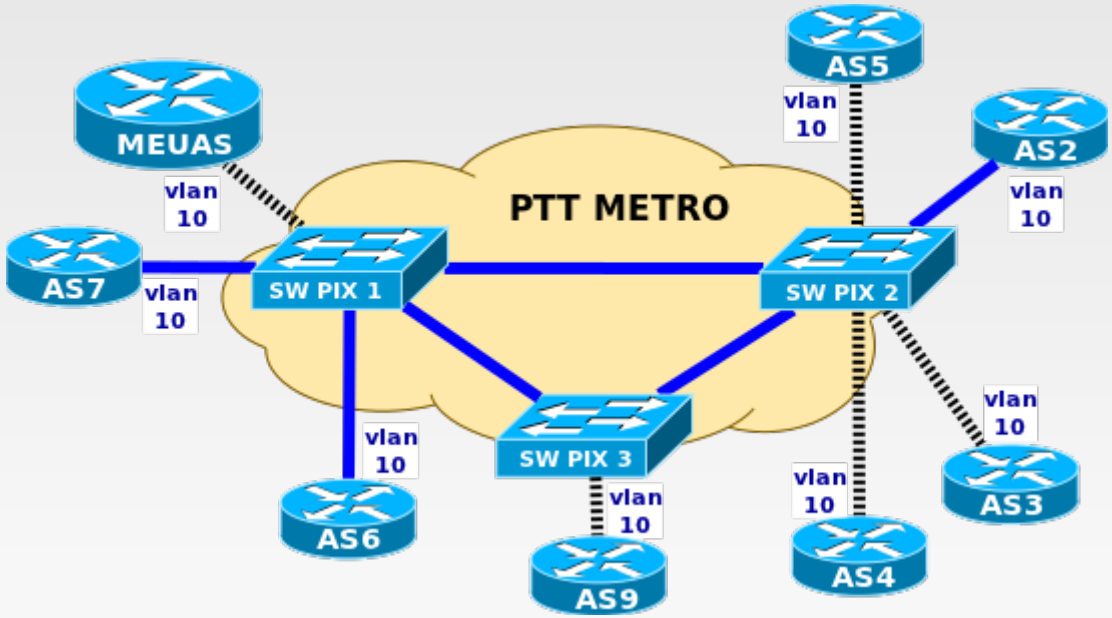
3.5
PARTICIPANTES LOCAIS E REMOTOS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.5 - Participantes locais e remotos

- Para participar é preciso ter conectividade layer2 com o switch do PTT-Metro. Para isso há duas formas:

  #1- Conectando-se fisicamente em um dos switches de um dos PIXs atraves de cabo/fibra

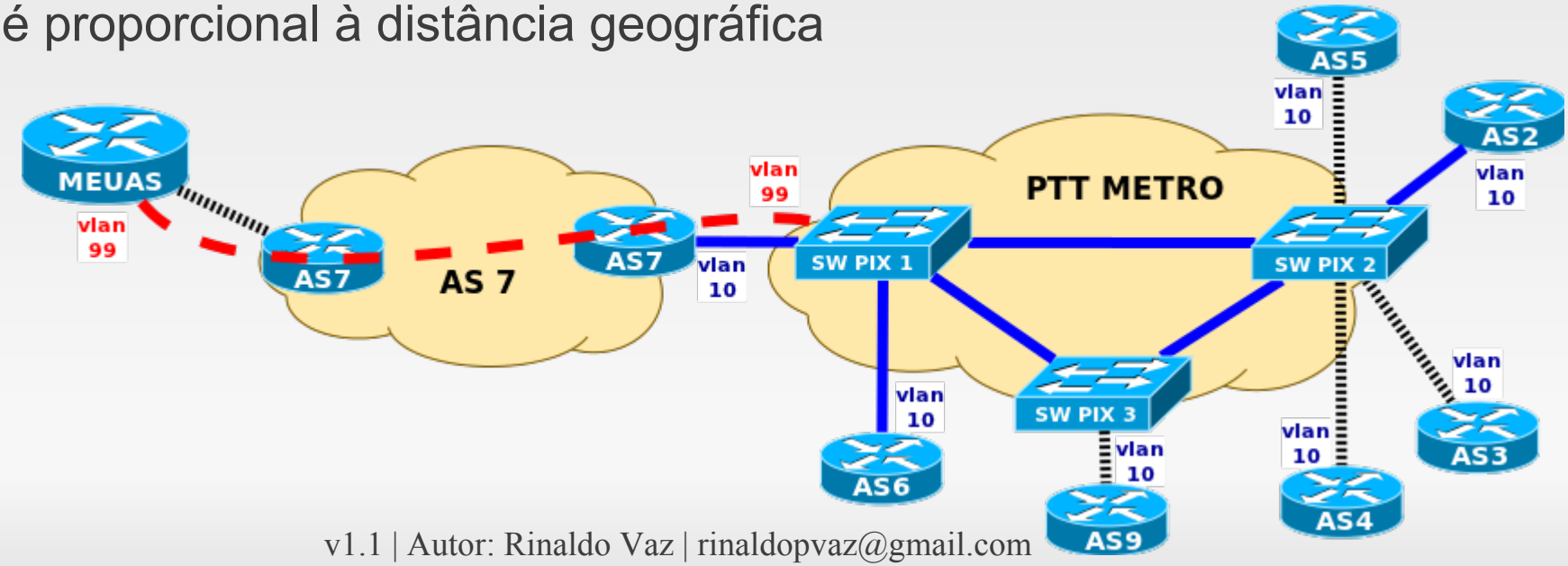


3.5 - Participantes locais e remotos

- Ou...

  2# - Contratando transporte layer2 (lan2lan, clear channel) de algum participante conectado fisicamente

- Não há diferenças no ponto de vista técnico entre as duas modalidades, exceto pelas consequências do aumento de latencia, que é proporcional à distância geográfica



3.6
ATM-TRATANDO AS ROTAS RECEBIDAS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.6 - Tratando as rotas recebidas

- LAB: Estabelecer sessões BGP com o PTT Route-server e com o PTT Looking Glass

```
GX-R1#conf t
GX-R1(config)#router bgp x
GX-R1(config-router)#neighbor 187.16.216.253 remote-as 555
GX-R1(config-router)#neighbor 187.16.216.253 description PTT-Route_server-1
GX-R1(config-router)#neighbor 187.16.216.253 weight 15
!--- o valor 15 de weight deve ser maior do que o aplicado no ”IN” do AS100, porém menor que o aplicado no peering "20"
GX-R1(config-router)#neighbor 187.16.216.253 flter-list 1 out
!--- vamos anunciar apenas o bloco /8 para o PTT e todos os outros peers

GX-R1(config-router)#neighbor 187.16.216.252 remote-as 555
GX-R1(config-router)#neighbor 187.16.216.252 description PTT-Looking Glass
!--- para o looking glass do PTT enviaremos todas as rotas da FIB, portando nao usaremos nenhum filtro
```

3.6 - Tratando as rotas recebidas

- Apenas com o objetivo de simplificar o LAB, anunciaremos nada mais que o classe A para trânsito IP e PTT

```
GX-R1#conf t
GX-R1(config)#no ip prefix-list MEUS-BLOCOS
!--- remove a prefix-list evitando inconsistências
GX-R1(config)#ip prefix-list MEUS-BLOCOS permit 2.0.0.0/8
!--- insere o classe A na prefix-list MEUS-BLOCOS

GX-R1(config)#router bgp X
!--- no modo de configuração BGP adicionaremos a prefix-list MEUS-BLOCOS para
!--- filtrar a saída de todos os neighbors BGP exceto para o looking glass e para o peering

GX-R1(config-router)#neighbor 187.16.216.253 prefix-list MEUS-BLOCOS out
!--- route server do PTT

GX-R1(config-router)#neighbor 200.X.1.1 prefix-list MEUS-BLOCOS out
!--- transito AS 200

GX-R1(config-router)#neighbor 100.X.1.1 prefix-list MEUS-BLOCOS out
!--- transito AS 100
GX-R1(config-router)#end
GX-R1#clear ip bgp * soft
```

3.6 - Tratando as rotas recebidas

- Agora vamos verificar quais rotas recebo do Route Server do PTT

  Utilizaremos o router do grupo 9 como exemplo:

```
G09-R1#sh ip bgp summary
...
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
9.10.1.2	4	10	442	443	168	0	0	07:16:05	4
100.9.1.1	4	100	582	441	168	0	0	07:14:37	35
187.16.216.252	4	555	10	44	168	0	0	00:06:02	0
187.16.216.253	4	555	14	8	168	0	0	00:02:34	45
200.9.1.1	4	200	548	441	168	0	0	07:14:26	34

- Vamos confirmar se todas as 5 sessões estão estabelecidas:
  - Peering
  - Trânsito AS100
  - PTT Looking Glass
  - PTT Route Server
  - Trânsito AS200

3.6 - Tratando as rotas recebidas

- Vamos verificar todas as opções para chegar ao AS 123

  Continuaremos Utilizando o router do grupo 9 como exemplo:

```
G09-R1#sh ip bgp 123.123.123.1
BGP routing table entry for 123.0.0.0/8, version 160
Paths: (3 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  187.16.216.252

123 (received & used)
    187.16.216.123 from 187.16.216.253 (187.16.216.253)
      Origin IGP, localpref 100, weight 15, valid, external, best

  200 222 123
    200.9.1.1 from 200.9.1.1 (200.200.4.2)
      Origin IGP, localpref 100, valid, external

  100 111 55 123
    100.9.1.1 from 100.9.1.1 (100.100.100.4)
      Origin IGP, localpref 100, weight 10, valid, external
```

- A rota escolhida para chegar ao AS 123 é pelo PTT

3.6 - Tratando as rotas recebidas

- Agora vamos entrar no Looking Glass do AS123 e verificar as rotas para o AS do grupo estão aparecendo assim:

```
AS-123-R1>sh ip bgp X.0.0.0/8
BGP routing table entry for X.0.0.0/8, version 251
Paths: (3 available, best #3, table Default-IP-Routing-Table, not advertised outside local AS)
  Not advertised to any peer

  55 111 100 X
    123.55.2.2 from 123.55.2.2 (55.55.55.2)
      Origin IGP, localpref 100, valid, external

  222 200 X
    123.222.2.2 from 123.222.2.2 (222.222.222.2)
      Origin IGP, localpref 100, valid, external

  X
    187.16.216.X from 187.16.216.253 (187.16.216.253)      !--- Essa rota foi recebida do PTT
      Origin IGP, localpref 100, valid, external, best
      Community: local-AS
```

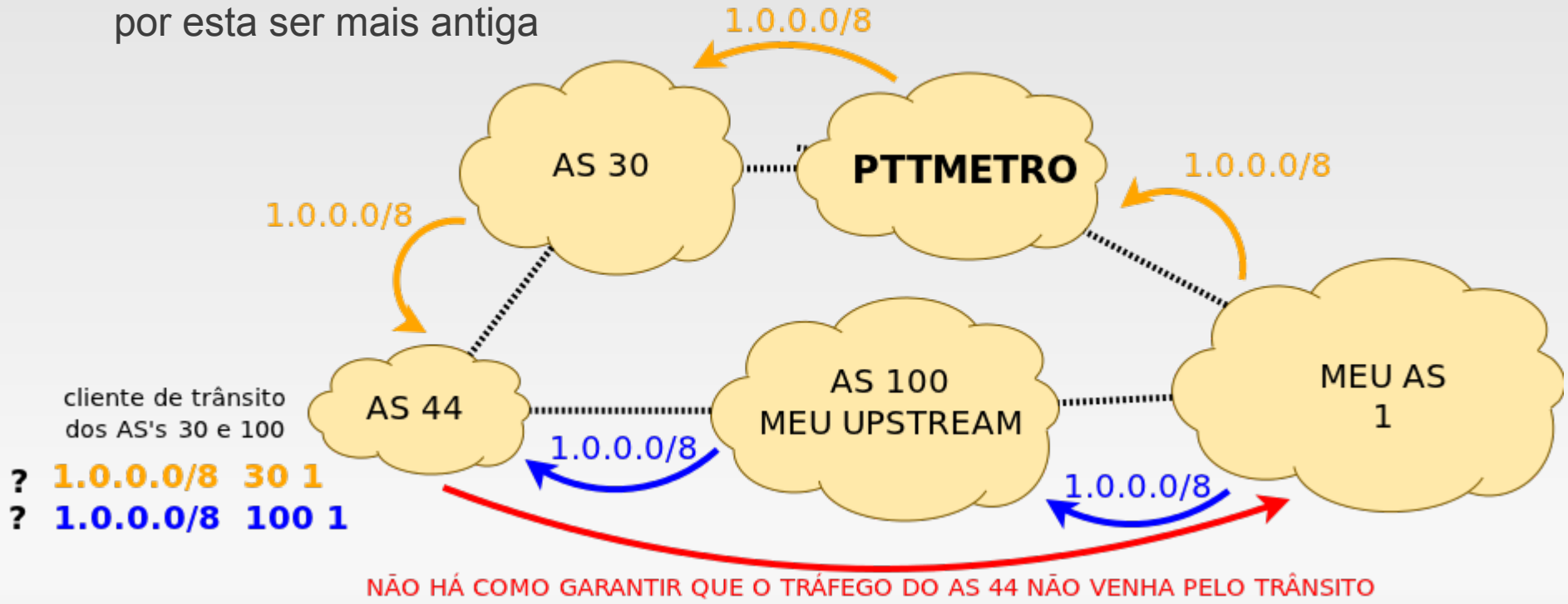
- A rota escolhida para chegar ao AS do grupo também é pelo PTT

3.7
ATM- ANUNCIANDO CORRETAMENTE NOS PONTOS DE TROCA

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

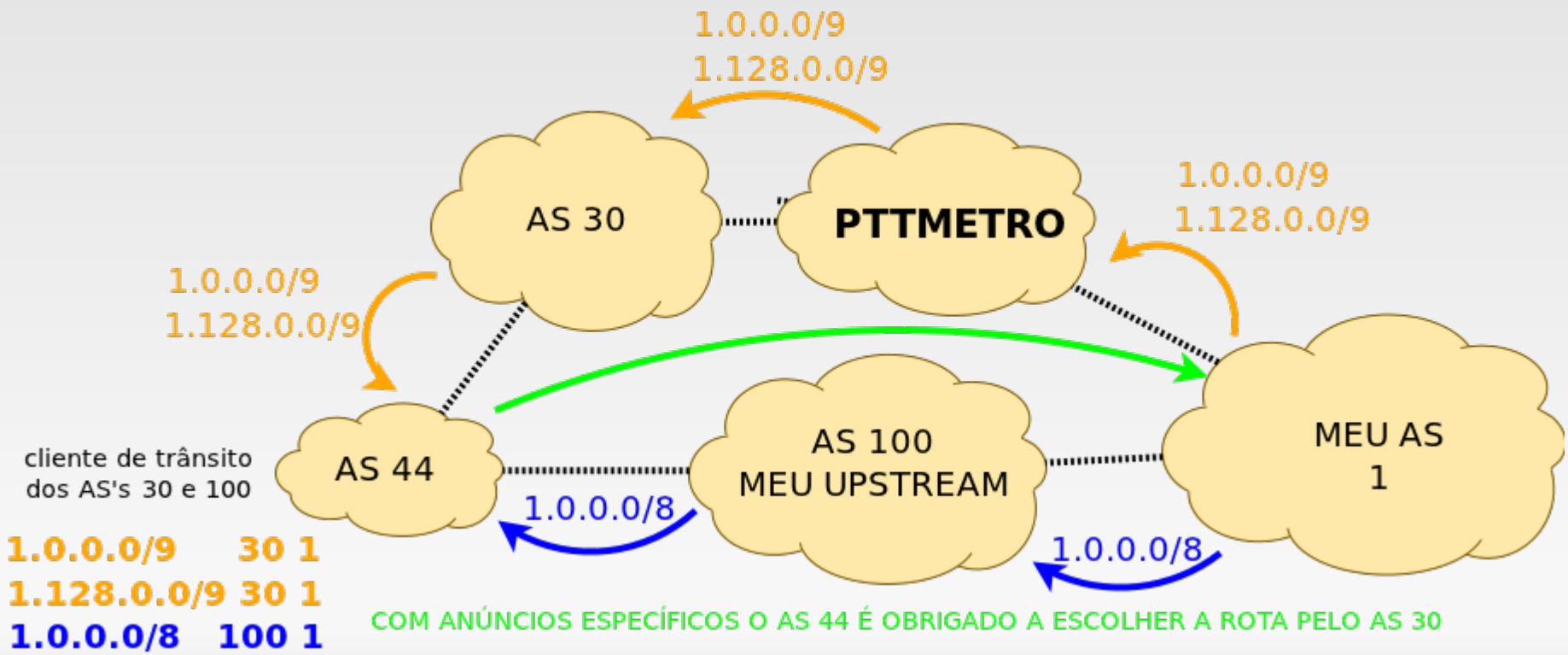
3.7- Anunciando corretamente nos pontos de troca

- Utilizamos anúncios da maneira mais simples possível, porém, no mundo real o mais recomendado é anunciar blocos mais específicos para os pontos de troca
- Um dos principais motivos são os participantes indiretos, digamos que o administrador do AS 44 coloca baixa prioridade para rotas recebidas pelo AS 30, ou mesmo que o BGP tenha escolhido sozinho o AS 100 como melhor rota por esta ser mais antiga



3.7- Anunciando corretamente nos pontos de troca

- Os anúncios garantem tráfego de download pelo PTT, porém existem outras maneiras de fazer isso, utilizando communities e as path prepend.



3.8
O LOOKING GLASS DO PTT-METRO

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.8 - O looking glass do PTT Metro

- O Looking Glass do PTT-Metro foge um pouco do padrão, quando visualizamos uma rota como ”best” não significa que essa é a rota que está sendo escolhida pelos participantes do PTT

- Ele não envia nenhuma rota, e todos os participantes devem enviar a sua tabela full

- Assim, é uma ferramenta para ajudar a visualizar toda a tabela de roteamento de cada participante

3.8 - O looking glass do PTT Metro

- Digamos que quero verificar a tabela de roteamento do participante do PTT que possui o AS 9

GX-R1>telnet 187.16.216.252
Trying 187.16.216.252 ... Open

User Access Verification

Password: 123

PTT-LG>show ip bgp regexp _9$

- Esse comando mostraria todas as rotas BGP na FIB do AS 9 (desde que esse estivesse enviando sua tabela full para o LG)

3.9
APLICAÇÕES PARA UM ACORDO BILATERAL

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

3.9 Aplicações para um acordo bilateral

- Com um acordo bilateral um AS pode vender trânsito para outro AS participante dispensando os custos de ativação (estrutura e equipamentos)

- Outra aplicação interessante é uma parceria entre 2 AS's de cidades diferentes para interligar um cliente comum em L2

3.9 Aplicações para um acordo bilateral

- LAB acordo bilateral:
  - Os grupos escolherão com quem fazer o acordo bilateral e terão uma vlan comum escolhida pelo instrutor
  - O objetivo é semelhante ao LAB do ”peering”, porém, um dos participantes fornecerá trânsito IP para o outro que será apenas cliente.
  - Para isso, deve ser estabelecida uma sessão BGP entre os dois, onde o responsável em prover trânsito deverá anunciar a tabela full para o cliente ao mesmo tempo que anuncia os prefixos do cliente para suas duas operadoras AS 100 e AS 200

3.9 Aplicações para um acordo bilateral

- Após decidido o ID da vlan cada participante deve inserir a tag correspondente na interface f0/1 (interface conectada ao PTT)
- Os IPs não importam para o PTT e devem ser combinados entre os dois participantes do acordo bilateral

G**X**-R1#conf t

G**X**-R1(config)#interface fastEthernet 0/1.**XXX**

G**X**-R1(config-subif)#encapsulation dot1Q **XXX**

G**X**-R1(config-subif)#ip address (combinem os endereços IP de cada lado)

G**X**-R1(config-subif)#end

G**X**-R1# ping **Z.Z.Z.Z**

- **XXX** é o número (ID) da vlan e **Z.Z.Z.Z** o endereço IP que o cliente configurou do outro lado

!--- testem conectividade IP através da vlan antes de iniciar as config BGP

3.9 Aplicações para um acordo bilateral

- Configurações BGP no router que será TRÂNSITO:
  - OBS: onde houver **Y** substituiremos pelo AS do CLIENTE, e **Z.Z.Z.Z** pelo IP que foi combinado previamente com o cliente e configurado do outro lado da VLAN

```
GX-R1#conf t
GX-R1(config)#router bgp X
GX-R1(config-router)#neighbor Z.Z.Z.Z remote-as Y
GX-R1(config-router)#neighbor Z.Z.Z.Z description Acordo bilateral - cliente
GX-R1(config-router)#neighbor Z.Z.Z.Z weight 25
!--- o valor de weight deve ser maior do todos os outros já que se trata de um cliente de trânsito
GX-R1(config-router)#neighbor Z.Z.Z.Z prefix-list BLOCOS-CLIENTE-AS-Y in
!--- dessa vez utilizaremos um filtro de entrada "in" para não haver risco de receber do cliente
!--- prefixos que não sejam  dele e meu tráfego de output ir na direção desse cliente
GX-R1(config-router)#exit

GX-R1(config)#ip prefix-list BLOCOS-CLIENTE-AS-Y permit Y.0.0.0/8
!--- adiciono o classe A do cliente na prefix-list declarada como filtro de entrada

GX-R1(config)#ip prefix-list MEUS-BLOCOS permit Y.0.0.0/8
!--- como já utilizamos a prefix-list "MEUS-BLOCOS" na saída dos AS de trânsito,
!--- vamos "aproveita-la" e apenas adicionar mais um prefixo
GX-R1(config-router)#end
GX-R1#clear ip bgp * soft
```

3.9 Aplicações para um acordo bilateral

- Configurações BGP no router CLIENTE:
  - OBS: onde houver **Y** substituiremos pelo AS que será TRÂNSITO, e **Z.Z.Z.Z** pelo IP que está configurado do outro lado

G**X**-R1#conf t
G**X**-R1(config)#router bgp **X**
G**X**-R1(config-router)#neighbor **Z.Z.Z.Z** remote-as **Y**
G**X**-R1(config-router)#neighbor **Z.Z.Z.Z** description Acordo bilateral-transito IP
G**X**-R1(config-router)#neighbor **Z.Z.Z.Z** prefix-list MEUS-BLOCOS out
!--- como cliente devemos anunciar da mesma maneira que fizemos com os ASs de trânsito
!--- e já que o classe A local já está nessa lista não preciso configurar mais nada
G**X**-R1(config-router)#end
*G**X**-R1#clear ip bgp * soft*

3.9 Aplicações para um acordo bilateral

- Testando o LAB:
  - Desativem as interfaces s1/0 e 1/1 do AS que foi designado como cliente no LAB
  - Se tudo deu certo, os ASs que não participam do PTT (55,100,111,200 e 222) conseguirão alcançar o AS do cliente através do AS que serviu como trânsito

4.1
UPSTREAMS E DOWNSTREAMS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.1 - Upstreams e downstreams

- Daqui por diante faremos referência à 3 tipos de sessões BGP:
  - IBGP (BGP interno): sessões BGP com routers do meu próprio AS
  - eBGP UPSTREAM: sessões BGP com ASs externos que prestam serviço de trânsito IP para meu AS
  - eBGP DOWNSTREAM: sessões BGP com ASs externos aos quais meu AS presta serviço de trânsito IP

4.2
CONFIGURANDO FILTROS DE UPSTREAMS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.2 - Configurando filtros de upstreams

- Não podemos anunciar nenhuma rede da qual não queremos receber DOWNLOADS, para isso utilizamos filtros em todas as nossas saídas de UPSTREAM e PTTs

- Porém existem certas precauções que devem ser tomadas nos filtros de **entrada** de UPSTREAM:
  - Não receber meus próprios prefixos;
  - Não receber BOGONS
  - Não receber IPs privados
  - Não receber rotas cujo AS PATH contenham AS(a) privado(a)
  - Garantir que rotas de UPSTREAMS nao recebam prioridade maior que rotas DOWNSTREAMS

4.2 - Configurando filtros de upstreams

- ”Por padrão” o BGP descarta todas as rotas que anunciei e voltaram para mim por algum motivo (multihoming)
- Quando isso acontece, logo é verificado o meu próprio AS no path e a rota é descartada automaticamente;
- Porém é possível que alguém anuncie por "engano", e a única forma de garantir é um filtro baseado em IP (prefix-list)

4.2 - Configurando filtros de upstreams

- Precisamos configurar alguns filtros padrão que evitam o recebimento de IPs privados, multcast, link local, etc
- Para isso vamos criar uma prefix-list chamada ”**NAO-RECEBER**” e inserir nela todas as redes indesejadas
- Para facilitar, o Cisco IOS permite especificar intervalos de IPs, isso dispensa o trabalho de inserir cada sub-rede manualmente
- Vamos entender melhor...

4.2 - Configurando filtros de upstreams

- Antes de criar uma prefix-list precisamos ter em mente que para toda ela há um DENY implícito no final, semelhante à uma ”lista branca”, onde quem estiver na lista será permitido, e todo o resto bloqueado, ex:

```
Gx-R1(config)#ip prefix-list MINHA-LISTA permit 122.36.200.0/24
Gx-R1(config)#ip prefix-list MINHA-LISTA {NEGAR TUDO}
```

- Com essa prefix-list na ENTRADA de um protocolo de roteamento qualquer o resultado seria receber apenas 122.36.200.0/24, e qualquer outro prefixo seria descartado
- Lembrem que a última regra de **{NEGAR TUDO}** jamais aparece, nem mesmo no ”show running-config”

4.2 - Configurando filtros de upstreams

- O que é o ”seq” ?
  - Ao inserir o comando:

```
Gx-R1(config)#ip prefix-list MINHA-LISTA permit 122.36.200.0/24
```

  - Automaticamente o IOS insere um número de sequência:

```
Gx-R1#show running-config | begin MINHA-LISTA
ip prefix-list MINHA-LISTA seq 5 permit 122.36.200.0/24
!
…
```

  - Caso adicionemos mais redes na mesma lista o número de seq é incrementado em 5 automaticamente para cada regra:

```
Gx-R1(config)#ip prefix-list MINHA-LISTA permit 44.0.0.0/24
Gx-R1(config)#end
Gx-R1#show running-config | begin MINHA-LISTA
ip prefix-list MINHA-LISTA seq 5 permit 122.36.200.0/24
ip prefix-list MINHA-LISTA seq 10 permit 44.0.0.0/24
!
...
```

4.2 - Configurando filtros de upstreams

- O que é o ”seq” ?
  - A razão para o incremento de ”5 em 5” é a possibilidade de inserir uma regra ”no meio” especificando um ”número de seq”:

```
Gx-R1(config)#ip prefix-list MINHA-LISTA seq 4 permit 10.0.30.0/24
```

  - A regra recém criada seria processada primeiro do que todas as outras:

```
Gx-R1#show running-config | begin MINHA-LISTA
ip prefix-list MINHA-LISTA seq 4 permit 10.0.30.0/24
ip prefix-list MINHA-LISTA seq 5 permit 122.36.200.0/24
ip prefix-list MINHA-LISTA seq 10 permit 44.0.0.0/24
!
…
```

  - Pouco importa a ordem de processamento quando todas as regras são de ”permit”
  - O ”seq” só tem relevância quando há regras de deny e permit na mesma prefix-list e/ou utilizamos ”ranges” com ”le/ge”

4.2 - Configurando filtros de upstreams

- Ranges de subredes - **(le)**:
  - Vamos agora considerar a necessidade de permitir sub-blocos:

```
Gx-R1(config)#ip prefix-list MINHA-LISTA permit 192.168.0.0/24
```

  - Essa regra permite 192.168.0.0/24, porém **não** permite 192.168.0.0**/25**, 192.168.0.128**/25**, 192.168.0.16**/28** ou qualquer outra sub-rede
  - Para incluir sub-redes utilizamos ”le”

```
Gx-R1(config)#ip prefix-list MINHA-LISTA permit 192.168.0.0/24 le 29
```

  - Com essa mudança, 192.168.0.0**/25**, 192.168.0.128**/25**, 192.168.0.16**/28** ou qualquer outra sub-rede dentro começando com **192.168.0** seria aceita, exceto redes **/30**, **/31** e **/32**
  - Para permitir TUDO dentro do /24 incluindo /30, /31 e /32 trocaríamos o **le 29 por le 32**

4.2 - Configurando filtros de upstreams

- Ranges de subredes - **(ge)**:
  - Diferente do **le**, o ge especifica o tamanho mínimo da máscara, seu uso é mais apropriado em redes maiores

    G**x**-R1(config)#ip prefix-list MINHA-LISTA permit 192.0.0.0**/8 ge 18 le 26**

  - O **/8** indica os bits que serão ”fixos”

  - Para esse exemplo, qualquer prefixo com os 8 primeiros bits fixos em começando com 192 (11000000) e ainda com máscara igual ou maior que **/18** e igual ou menor que **/26**

  - Caso seja declarada uma máscara /0 (192.0.0.0**/0**) não importariam os 8 primeiros bits e qualquer prefixo (começando com qualquer coisa) seria aceito desde que sua máscara fosse entre /18 e /26

4.2 - Configurando filtros de upstreams

- Ranges de subredes - **(ge)**:
  - Quero aceitar qualquer prefixo /24 e descartar todo o resto:

```
Gx-R1(config)#ip prefix-list MINHA-LISTA permit 0.0.0.0/0 ge 24 le 24
```

  - Quero aceitar TODOS os prefixos dentro de 200.233.16.0/20, menos o primeiro /22:

```
Gx-R1(config)#ip prefix-list MINHA-LISTA seq 10 deny 200.233.16.0/22
Gx-R1(config)#ip prefix-list MINHA-LISTA seq 20 permit 200.233.16.0/20 le 32
!--- na vida real ninguem recebe prefixos menores que /24 por BGP,
!--- utilizaremos no máximo "le 24" para qualquer filtro (in – out)

!--- se posteriormente fosse necessário adicionar bloqueio de mais um /22,
!--- o bloqueio funcionaria apenas com um seq menor que "20"
```

4.2 - Configurando filtros de upstreams

- Agora que compreendemos o funcionamento, vamos continuar de onde paramos e criar uma prefix-list chamada **NAO-RECEBER**
- Dentro dela colocaremos todos os prefixos que não desejamos receber
- Por se tratar de uma ”lista negra” não podemos esquecer de inserir uma regra que permita todo o resto no fim da hierarquia

4.2 - Configurando filtros de upstreams

- Inserindo os prefixos na lista:

```
GX-R1(config)#ip prefix-list NAO-RECEBER seq 10 deny X.0.0.0/8 le 32
!--- blocos do meu próprio AS
GX-R1(config)#ip prefix-list NAO-RECEBER seq 20 deny 10.0.0.0/8 le 32
!--- uso privado (neste lab estamos utilizando para grupo 10, favor nao bloquear este)
GX-R1(config)#ip prefix-list NAO-RECEBER seq 30 deny 172.16.0.0/12 le 32
!--- uso privado
GX-R1(config)#ip prefix-list NAO-RECEBER seq 40 deny 169.254.0.0/16 le 32
!--- link-local (APIPA)
GX-R1(config)#ip prefix-list NAO-RECEBER seq 50 deny 127.0.0.0/8 le 32
!--- loopbacks
GX-R-1(config)#ip prefix-list NAO-RECEBER seq 60 deny 192.0.2.0/24 le 32
!--- Documentação
GX-R1(config)#ip prefix-list NAO-RECEBER seq 70 deny 192.168.0.0/16 le 32
!--- uso privado
GX-R1(config)#ip prefix-list NAO-RECEBER seq 80 deny 224.0.0.0/4 le 32
!--- multicast
GX-R1(config)#ip prefix-list NAO-RECEBER seq 90 deny 240.0.0.0/4 le 32
!--- reservado para uso futuro (para o fim do IPv6?)
GX-R1(config)#ip prefix-list NAO-RECEBER seq 500 permit 0.0.0.0/0 le 24
!--- Permitir todo o resto e ao mesmo tempo descartar qualquer prefixo
!--- mais específico que /24 como /25, /26, /27, etc
```

4.2 - Configurando filtros de upstreams

- Apenas criar a lista **NAO-RECEBER** de nada vale se a mesma não for declarada como filtro de ENTRADA dos UPSTREAMS, PEERING e PTT

```
GX-R1(config)#router bgp X
GX-R1(config-router)#neighbor 100.x.1.1 prefix-list NAO-RECEBER in
GX-R1(config-router)#neighbor 200.x.1.1 prefix-list NAO-RECEBER in
GX-R1(config-router)#neighbor 187.16.216.253  prefix-list NAO-RECEBER in
GX-R1(config-router)#neighbor (IP de peering) prefix-list NAO-RECEBER in
GX-R1(config-router)#neighbor (acordo bilateral) prefix-list NAO-RECEBER in
GX-R1(config-router)#end
GX-R1#clear ip bgp * soft
```

4.2 - Configurando filtros de upstreams

- Desmistificando o ”Soft reconfiguration inbound”:

  - Para que serve?

4.2 - Configurando filtros de upstreams

- HARD reset:
  - É a maneira menos recomendada de fazer um refresh nas rotas
  - A conexão TCP volta para o estado ”idle”
  - Há um pequeno intervalo antes da conexão iniciar novamente

    GX-R1#clear ip bgp { * | A.B.C.D | peergroupname }
- SOFT reset:
  - Realiza o route refresh mantendo a sessão ativa
  - Pode ser dividido em ”outbound refresh” e ”inbound refresh”
    - GX-R1#clear ip bgp { * | A.B.C.D | peergroupname } soft { out | in }
    -
- Porém isso ainda não explica a função do ”soft reconfiguration inbound” ...

4.2 - Configurando filtros de upstreams

- Quando updates são bloqueados pelos filtros de ”input”, esses updates são descartados pelo router, não sendo possível verificar informações sobre essas rotas

- Habilitar o ”soft reconfiguration inbound” para um neighbor reserva uma área em memória para guardar informações de todas as rotas recebidas desse neighbor, tornando possível verificar as rotas que foram descardadas pelos filtros de ”input”

- Habilitar o ”soft reconfiguration inbound” permitirá usar o comando:

```
GX-R1#sh ip bgp neighbors 200.9.1.1 received-routes
    !--- exibe as rotas recebidas de 200.9.1.1 antes dos filtros de input
    !--- inclusive rotas com next-hop inválido

GX-R1#sh ip bgp neighbors 200.9.1.1 routes
    !--- exibe as rotas recebidas de 200.9.1.1 após os filtros de input
```

4.2 - Configurando filtros de upstreams

- Vamos habilitar o ”Soft reconfiguration inbound” para todos os neighbors:

```
GX-R1(config)#router bgp X
GX-R1(config-router)#neighbor 100.X.1.1 soft-reconfiguration inbound
GX-R1(config-router)#neighbor 200.X.1.1 soft-reconfiguration inbound
GX-R1(config-router)#neighbor 187.16.216.253  soft-reconfiguration inbound
GX-R1(config-router)#neighbor (IP de peering) soft-reconfiguration inbound
GX-R1(config-router)#neighbor (acordo bilateral) soft-reconfiguration inbound
GX-R1(config-router)#end
GX-R1#clear ip bgp * soft
```

4.2 - Configurando filtros de upstreams

- O AS 123 vai anunciar vários IPs privados, vamos verificar se o bloqueio está acontecendo corretamente:
  - Primeiro pelo AS 200

```
GX-R1#sh ip bgp neighbors 200.9.1.1 received-routes
    !--- todas as rotas ANTES do filtro "NAO-RECEBER"
GX-R1#sh ip bgp neighbors 200.9.1.1 routes
    !--- todas as rotas APÓS do filtro "NAO-RECEBER"
```

  - Depois pelo AS 100

```
GX-R1#sh ip bgp neighbors 200.9.1.1 received-routes
    !--- todas as rotas ANTES do filtro "NAO-RECEBER"
GX-R1#sh ip bgp neighbors 200.9.1.1 routes
    !--- todas as rotas APÓS do filtro "NAO-RECEBER"
```

4.2 - Configurando filtros de upstreams

- Outros BOGONS
  - Existe outro range de IPs que não são privados, mas ainda assim devem ser bloqueados
  - Existem blocos públicos de IPs que ainda não foram designados pelas organizações responsáveis.
  - Dessa forma é possível que alguém utlize essas redes com intenções suspeitas
  - Onde conseguir uma lista atualizada?

4.2- Configurando filtros de upstreams

- Melhor do que sair à procura de listas é estabelecer um peering com o cymru

- Para solicitar esse peering é preciso entrar em contato com bogonrs@cymru.com com as seguintes informações:

  - Número de AS
  - Ip(s) que o cymru fará peering com você
  - Se o equipamento suporta autenticação MD5 para BGP
  - Chave pública GPG/PGP (opcional)

4.2- Configurando filtros de upstreams

- Segundo o site a resposta com as informações chega em 2 dias úteis em média

- Para mais informações acesse:
  http://www.team-cymru.org/Services/Bogons/bgp.html

4.2 - Configurando filtros de upstreams

- Configurando a sessão BGP com Cymru...

GX-R1(config)#router bgp X

GX-R1(config-router)#neighbor x.x.x.x remote-as 65333

GX-R1(config-router)#neighbor x.x.x.x ebgp-multihop 255

GX-R1(config-router)#neighbor x.x.x.x weight 100

GX-R1(config-router)#neighbor x.x.x.x description BOGONS CYMRU

GX-R1(config-router)#neighbor x.x.x.x prefix-list SAIDA-CYMRU out

GX-R1(config-router)#neighbor x.x.x.x route-map FILTRO-CYMRUBOGONS in

GX-R1(config-router)#neighbor x.x.x.x password <senha cymru>

GX-R1(config-router)#exit

4.2 - Configurando filtros de upstreams

- Adicionando as rotas recebidas em **BLACKHOLE**

```
GX-R1(config)#ip bgp-community new-format
!---habilita suporte à communities tipo xxxx:yyyy
GX-R1(config)#ip route 192.0.2.1 255.255.255.255 null0
!---cria a rota de blackhole
GX-R1(config)#ip community-list 10 permit 65333:888
!---adiciona a community 65333:888 na comm-list 10
GX-R1(config)#route-map FILTRO-CYMRUBOGONS permit 10
!---entra as opções do route-map de entrada
GX-R1(config-route-map)#match community 10
!---declara o conteudo da comm-list 10 para condição de math
GX-R1(config-route-map)#set ip next-hop 192.0.2.1
!---para toda rota que fizer a condição mudar o next-hop para blackhole
GX-R1(config)#ip prefix-list SAIDA-CYMRU seq 5 deny 0.0.0.0/0 le 32
!---garantir que nada será anunciado para o cymru
```

4.3
CONFIGURANDO FILTROS DOWNSTREAMS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.3 - Configurando filtros de downstreams

- A primeira coisa e se levar em conta é a prioridade maior para os prefixos recebidos do meu downstream

- Ao mesmo tempo, não podemos correr o risco de receber um full-routing desse cliente por algum erro dele, caso isso aconteça. todo meu upload passa a subir por esse cliente.

4.3 - Configurando filtros de downstreams

- LAB: Há um cliente no PTT que vai comprar trânsito IP de todos os alunos

- Informações:
  - AS: 99
  - Bloco CIDR: 99.0.0.0/8
  - IP: 187.16.216.99

4.3 - Configurando filtros de downstreams

- Como já usamos os pesos (weight) 10 15 e 20 utilizaremos 30 para downstreams

```
GX-R1(config-router)#neighbor 187.16.216.99 remote-as 99
GX-R1(config-router)#neighbor 187.16.216.99 weight 30
GX-R1(config-router)#neighbor 187.16.216.99 prefix-list BLOCOS-AS-99 in
!--- filtro de ENTRADA para o DOWNSTREAM AS 99
GX-R1(config-router)#neighbor 100.X.1.1 prefix-list ANUNCIAR-AS-100 out
!--- filtro de SAIDA para o UPSTREAM AS 100
GX-R1(config-router)#neighbor 200.X.1.1 prefix-list ANUNCIAR-AS-200 out
!--- filtro de SAIDA para o UPSTREAM AS 200
GX-R1(config-router)#exit
GX-R1(config)#ip prefix-list BLOCOS-AS-99 permit 99.0.0.0/8 le 24
!--- como declaramos a prefix-list BLOCOS-AS-99 na entrada do DOWNSTREAM vamos adicionar
!--- os blocos CIDR do DOWNSTREAM nessa lista
GX-R1(config)#ip prefix-list ANUNCIAR-AS-100 permit 99.0.0.0/8 le 24
GX-R1(config)#ip prefix-list ANUNCIAR-AS-100 permit X.0.0.0/8 le 24
!--- adicionar os blcos CIDR do DOWNSTREAM e também os blocos do AS
!--- na saída do UPSTREAM AS 100
GX-R1(config)#ip prefix-list ANUNCIAR-AS-200 permit 99.0.0.0/8 le 24
GX-R1(config)#ip prefix-list ANUNCIAR-AS-200 permit X.0.0.0/8 le 24
!--- adicionar os blcos CIDR do DOWNSTREAM e também os blocos do AS
!--- na saída do UPSTREAM AS 100
```

4.4
ENTENDENDO O ”BGP REGEXP” (expressões regulares)

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.4 - Entendendo o regexp

- Quando temos muitos clientes de trânsito é preciso estabelecer padrões para que os administradores sejam capazes de entender as configurações

- Além de evitar ao máximo a quantidade excessiva de linhas de configuração

- Por exemplo:

  Há um "range" de números de AS que são de uso privado semelhantes aos IPs 192.168.x.x . Esse intervalo vai de 64512 até 65535, e não devemos receber anúncios que contenham algum desses AS's no path

4.4 - Entendendo o regexp

- Imagine ter que configurar uma linha de deny para cada AS privado

```
AS-X-R1(config)#ip as-path access-list 1 deny 64512
AS-X-R1(config)#ip as-path access-list 1 deny 64513
AS-X-R1(config)#ip as-path access-list 1 deny 64514
...
...
AS-X-R1(config)#ip as-path access-list 1 deny 65534
AS-X-R1(config)#ip as-path access-list 1 deny 65535
AS-X-R1(config)#ip as-path access-list 1 permit .*
```

- Seria um tanto complicado
- Felizmente com expressões regulares ao invés de milhares, resolvo esse problema com apenas 2, (ou uma talvez)

4.4 - Entendendo o regexp

- Antes vamos entender a função dos principais caracteres nas expressões regulares:

"_" O caracter "underline" define um espaço em branco

- **"1" Um número isolado satisfaz a condição para qualquer expressão que contenha esse número**

- **exemplo1**: **1** "a condição de match é qualquer AS no Path (podem haver vários AS's) que contenha o número 1 em qualquer parte, em pelo menos um dos AS's

  ***Satisfaz**:* ”99999999**1**9999999999”, "**1**23 432", "999 888 777 **1**23 998"

  N*ão satisfaz*: "355 3344 22334", "466", etc

- **exemplo2**: **_1** "a condição de match é qualquer AS no Path (podem haver vários AS's) que contenha o número 1 no ínicio de pelo menos um dos AS's e que essa ocorrência não seja a primeira”

Satisfaz: "**1**2333 3445 9999", "**1**23 432", "3333 **1**111 2222 4444"

N*ão satisfaz*: "4333 21123", "4661" "21111 12222", etc

- **exemplo3**: **_1_** "a condição de match é qualquer AS no Path (podem haver vários AS's) que contenha o número 1 sozinho em pelo menos um dos AS's e que essa ocorrência não seja nem a primeira nem o última

- **Satisfaz**: "2333 **1** 3445 9999", "33 **1** 65554", "433 4444 **1** 433"

Não satisfaz: "333 222 555 1"

4.4 - Entendendo o regexp

- **”.” Satisfaz a condição para qualquer número no lugar do ponto**

- **exemplo**: _**4.5**_ "a condição de match é qualquer AS PATH que contenha um AS de 3 dígitos onde 4 seja o primeiro e 5 seja o último com qualquer número no meio e que contenha outros AS's antes e depois

  Satisfaz: "12 **415** 43", "2 **425** 3", "6 **435** 2","321 456 **445** 345"

  Não satisfaz: "415", "123 415", etc

- **Exemplo**: **4.5** "a condição de match é qualquer número de 3 ou mais dígitos que fique "entre" 4 e 5 em qualquer parte do AS PATH

  Satisfaz: "23**415**", "43**425**", "**435** 3333", "**445**", "**455**87", "321 4**445**9 345"

  Não satisfaz: 315, 416, etc

4.4 - Entendendo o regexp

- ”**$**” Satisfaz a condição para o tudo que o que for terminado com o caracter (ou expressão) ao lado esquerdo de "$"
- **exemplo1**: **12$** "a condição de match é qualquer AS PATH em que os dois últimos dígitos sejam 1 e 2 respectivamente"

  Satisfaz: ”49**12**”, "650**12**", "123**12**", "111**12**", "9932 9976 649**12**", "28135 17379 543**12**" ...

  Não satisfaz: 13, 22, 129, 55413, 58752 …
- **exemplo2**: **_12**$ "a condição de math é qualquer AS PATH em que último AS seja 12"

  Satisfaz: "**12**", "2335 **12**", "312 **12**", "111 **12**"

  Não satisfaz: "444 112", "22", "12 229", "999 12 1111"
- **exemplo3**: **1.2**$ "a condição de match é qualquer AS PATH em que seus últimos 3 dígitos do último AS sejam: 1, qualquer coisa e 2 respectivamente"

  Satisfaz: "123 45666 33 **112**", "444 122", "**132**", "1111 **132**", "444 651**92**", "601**92**"

  Não satisfaz: "2333 222", 123, 912

4.4 - Entendendo o regexp

- ”**^**” Satisfaz a condição para caracteres que comecem com o caracter (ou exprssão) após o ^
- **exemplo1**: **^1** "a condição de match é qualquer AS PATH que tenha o dígito 1 no início do primeiro AS do PATH

  satisfaz: "**1**111 9999 55443", "**1**2345 6776", **1**2222, **1**9999

  não satisfaz: 21111, 31211, 2, 3, 43111
- **exemplo2**: **^1.4** "a condição de match é qualquer AS PATH em que os 3 primeiros dígitos do primeiro as sejam: 1, qualquer coisa e 4 respectivamente

  Satisfaz: "**124** 4321", "**124**44", "**14**49", "**19**455 12333", "**104**66 55443 9900"

  Não satisfaz: 195, 2124, 22444, 9104
- **exemplo3**: **^1..8$** "a condição de match é um AS PATH contento apenas um AS de 4 dígitos, onde o primeiro dígito é 1, o segundo e o terceiro qualquer coisa e o quarto e último seja 8 respectivamente

  Satisfaz: ”**1008**”, ”**1248**”, ”**1118**”, ”**1228**”, ”**1998**”, ”**1888**”

  Não satisfaz: 10008, "1008 1008", "14448"

4.4 - Entendendo o regexp

- ”*****” Satisfaz a condição para qualquer caractere à esquerda do "*", inclusive nenhum, por isso é preciso utilizar o caracter um espaço "_" branco em conjunto, pois sem espaço a expressão perde o sentido, por exemplo, 1* satisfaria condição para qualquer coisa, e o padrão intuitivo para "qualquer coisa/tudo" é ".*"

- **exemplo1**: **1**_* "a condição de match é qualquer coisa que contenha o dígito 1 no final de pelo menos um AS"

  satisfaz: "4343**1** 9999 777 9999 8888(infinito...)”,” … 9999 999**1** 345 666 654 5556...”

  não satisfaz: 8872, "9987 2332", etc

- **exemplo2**: **.*** "a condição de math é qualquer AS PATH de qualquer tamanho, com qualquer combinação"

  Satisfaz: TUDO

  Não satisfaz: NADA

4.4 - Entendendo o regexp

- ”\” Este caracter é usado quando os caracteres de condição também são caracteres de expressão regular, por exemplo, se precisássemos de uma expressão para dinheiro, onde a condição fosse "R$ 10,01" até "R$ 10,99", teríamos um grande problema, já que a expressão necessária é "R$_10,..$" Notem que o "$" precisa aparacer como "condição" ao mesmo tempo que é um caracter de expressão regular. Sendo assim, não funcionaria. O caracter "\" é usado para declarar que o próximo caracter deverá ser interpretado como "condição", ao invés de sua função normal. Assim, a expressão corretá ficaria "R\$_10,..$"
- ”Felizmente” não utilizaremos esse caracter para BGP

4.4 - Entendendo o regexp

- ”[ ]” Satifaz a condição para qualquer um dos caracters dentro dos colchetes, mas apenas UM deles. Há a possibilidade de utilizar intervalos entre os colchetes seguidos de hífem "-"
- Dentro dos colchetes o caracter "^" tem função de negação, O intervalo funciona da mesma forma para as negações.
- **exemplo1**: **^9[249]5$** "a condição é um AS que contenha qualquer combinação de 3 dígitos sendo o primeiro 9, o terceiro 5 e o do meio 2, 4 ou 9

  Satisfaz: ”**925**”, ”**945**”, e ”**995**”

  Não satisfaz: 915, 955, 965, etc
- **exemplo2**: **^9[5-7]5$** "mesma lógica do exemplo 1, porém o dígito do meio pode ser de 5 à 7"

  satisfaz: **955**, **965** e **975**

- **exemplo3**: **^9[^28]5$** "a condição de match é a mesma lógica dos exemplos 1 e 2 porém o digito do meio não poderá ser 2 ou 8

  satistaz: todas as possibilidades de um ”9.5” "menos" **925** e **985**.
- **exemplo4**: **^9[^5-8]5$** "a condição de match segue a mesma lógica dos exemplos 2 e 3 porém o dígito do meio não pode estar no intervalo entre 5 e 8 (5,6,7 ou 8)

  satisfaz todas as possibilidades "menos" **955**, **965**, **975** e **985**

4.4 - Entendendo o regexp

- ”|” O PIPE divide dois ou mais caracters em uma condiço "ou".

- **exemplo1**: **3(4|9)3** satisfaz: **343** ou **393** apenas

- **exemplo2**: **5([0-3]|9)3** satisfaz: **503**, **513**, **523**, **533** e **593**

- PS ( ) coloca uma expressão em evidência, sem eles o exemplo 2 aceitaria
  5333333333333333333353333333333333333333333435465769733...

4.4 - Entendendo o regexp

- ”+” Satisfaz a condição para pelo menos uma ou mais ocorrências do caracter (ou expressão) anterior
- **exemplo1**: **2+** satisfaz: 2, 22, 22222, 222222, 2222222222 ....

- ? Satisfaz a condição para uma ou nenhuma ou uma ocorrencia do caracter (ou expressão) anterior
- **exemplo1**: **2?** satisfaz: **2** ou " "

- ”{ }” especifica um número ou um intervalo de repetições para um caracter (ou expressão) antecedente
- **exemplo1**: **^2{4}$** satisfaz: **2222** apenas
- **exemplo2**: **^2{4,6}$** satisfaz: **2222**, **22222** e **222222**
- **exemplo3**: **^1[0-3]{1,2}$** satisfaz: **100**, **13**, **10**, **132**, **12**, **130**

  nao satisfaz: 14, 1031, 15, 1033

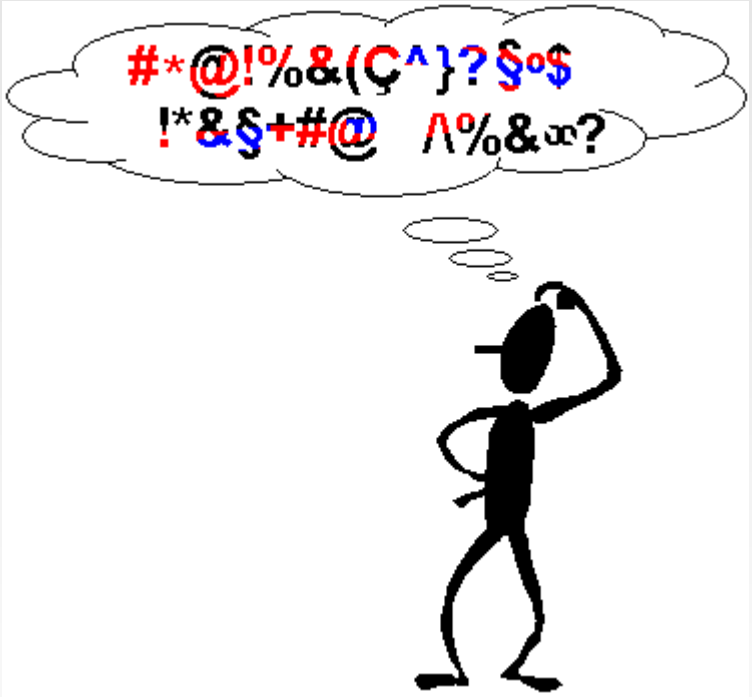
- Para praticar segue um link interessante:
- http://tools.lymas.com.br/regexp_br.php

4.5
Filtros facilmente escaláveis

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.5 - Filtros facilmente escaláveis

- Agora vamos raciocinar um pouco e utilizar as expressões que aprendemos para bloquear o range dos AS's 64512 até 65535 utilizando o menor número possível de regras...



4.5 - Filtros facilmente escaláveis

AS-**X**-R1(config)#ip as-path access-list 1 permit 645(0.|10|11)
!--- aceita de 64501 a 64511
AS-**X**-R1(config)#ip as-path access-list 1 permit ^……$
AS-**X**-R1(config)#ip as-path access-list 1 deny 6(4[5-9]..|5...)
!--- rejeita qualquer coisa de 64500 à 64999 e de 65001 à 65999
!--- obs, não há AS maior que 65535
AS-**X**-R1(config)#ip as-path access-list 1 permit .*
!--- aceita todo o resto

LAB:
O AS 99 vai gerar rotas com AS's privados, aplicar filtros de input para bloquear essas rotas:

AS-**X**-R1(config)#router bgp **X**
!--- Vamos adicionar o mesmo prefix-list na entrada do AS 100
AS-**X**-R1(config-router)#neighbor 100.**X**.1.1 filter-list 1 in

4.5 - Filtros facilmente escaláveis

- Existem infinitas aplicações, por exemplo, descubro que o AS 12345 está com lentidão, crio políticas que diminuem a prioridade de qualquer prefixo anunciado por esse AS

```
router(config)#ip as-path access-list 20 permit (_12345_|_12345$)
!--- cria a ACL 20 e diz que qualquer AS PATH contendo 12345 está nela

router(config)#route-map UPSTREAM-1 permit 1
!--- Cria um route-map no topo da hierarquia
router(config-route-map)#set local-preference 90
!--- especifica a ação "mudar o local pref para 90"
router(config-route-map)#match as-path 20
!--- especifica quem irá receber a ação "set local pref 90"
router(config-route-map)#exit
AS-x-R1(config)#route-map UPSTREAM-1 permit 2
!--- cria um route-map ”em branco”que aceita todo o resto sem modificações
router(config-route-map)#exit
```

4.5 - Filtros facilmente escaláveis

- Precisamos nesse caso fazer o mesmo para todos os UPSTREAMS

```
router(config)#route-map UPSTREAM-2 permit 1
!--- Cria um route-map no topo da hierarquia
router(config-route-map)#set local-preference 90
!--- especifica a ação "mudar o local pref para 90"
router(config-route-map)#match as-path 20
!--- especifica quem irá receber a ação "set local pref 90"
router(config-route-map)#exit
AS-x-R1(config)#route-map UPSTREAM-2 permit 2
!--- cria um route-map ”em branco”que aceita todo o resto sem modificações
router(config-route-map)#exit
```

4.5 - Filtros facilmente escaláveis

- As expressões regulares não servem apenas para filtrar prefixos, mas também para encontrar certas rotas entre as centenas de milhares de uma tabela full
- Por exemplo:
- Quero saber quais rotas na minha RIB foram originadas pelo AS 6762 (TELECOM ITALIA SPARKLE S.p.A.)

```
BGPD-ALOG-SP# sh ip bgp regexp _6762$
...
   Network          Next Hop            Metric LocPrf Weight Path
*> 2.16.70.0/23     201.20.36.158                            0 16397 6762 i
*> 2.16.140.0/23    201.20.36.158                            0 16397 6762 i
*> 2.16.146.0/23    201.20.36.158                            0 16397 6762 i
...
```

4.5 - Filtros facilmente escaláveis

- Exemplo 2:

Quais rotas foram recebidas pela Intelig através do peering com Embratel?

```
BGPD-ALOG-SP# sh ip bgp regexp _17379 4230_
...
   Network          Next Hop            Metric LocPrf Weight Path
*>  186.205.244.0/22 201.20.36.158                         0 16397 17379 4230 28573 i
```

- Exemplo 3:

Quais ASs estão diretamente conectados com o AS 4230 (EMBRATEL)?

```
BGPD-ALOG-SP# sh ip bgp regexp _4230 ([0-9]+){1}$
…
   Network          Next Hop            Metric LocPrf Weight Path
*>  23.61.118.0/23   201.20.36.158                         0 16397 17379 4230 28573 i
*>  23.61.120.0/22   201.20.36.158                         0 16397 17379 4230 28573 i
…
```

4.5 - Filtros facilmente escaláveis

- Quais ASs estão diretamente conectados com meus upstreams/downstreams?

```
BGPD-ALOG-SP# sh ip bgp regexp ^([0-9]+_){2}$
   Network          Next Hop            Metric LocPrf Weight Path
...
*>  216.72.166.0       201.20.36.157                          0 16397 6505 i
*>  216.239.32.0/19    201.20.36.158                          0 16397 15169 i
...
```

4.6
ENVIANDO/RECEBENDO PARCIAL E FULL-ROUTING

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.6 - Enviando/recebendo partial e full-routing

- É um conceito muito discutido sem conclusões, há quem diga que um "parcial" precisa ter um número específico
- Se a tabela full tem hoje 400.000 prefixos, 399.999 é um "parcial" assim como 1 prefixo também
- Quando se tem apenas uma operadora, de nada vai adiantar receber qualquer coisa além da rota padrão
- Com 2 ou mais, o parcial ideal é receber os DOWNSTREAMS dessas operadoras, e também os "DOWNSTREAMS desses DOWNSTREAMS"

4.6 - Enviando/recebendo partial e full-routing

- **Recebendo parcial routing do meu UPSTREAM:**

  Vamos configurar um exemplo de receber do meu AS 100 apenas os seus DOWNSTREAMS diretos e peerings

  G**X**-R1#conf t

  G**X**-R1(config)#router bgp **X**

  G**X**-R1(config-router)#neighbor 100.**X**.1.1 filter-list 2 in
  !--- declara a filter-list 2 para filtrar o input do UPSTREAM AS 100
  !--- Infelizmente o Cisco não aceita um NOME para uma as-path acl, no Quagga sim

  G**X**-R1(config)#ip as-path access-list 2 permit ^([0-9]+|[0-9]+ [0-9]+)$
  !--- aceita rotas com um ou doiss ASs no PATH

  G**X**-R1(config)#ip as-path access-list 2 deny .*
  !--- rejeita qualquer outro tamanho de AS PATH

- Mesma lógica para 3 as's no path e assim por diante

  ^([0-9]+|[0-9]+ [0-9]+|[0-9]+ [0-9]+ [0-9]+)$

4.6 - Enviando/recebendo partial e full-routing

- Vamos ver se funcionou...

G**X**-R1#sh ip bgp neighbors 200.9.1.1 received-routes

G**X**-R1#sh ip bgp neighbors 200.9.1.1 routes

*G**X**-R1#sh ip bgp neighbors 200.9.1.1 received-routes*

G**X**-R1#sh ip bgp neighbors 200.9.1.1 routes

4.6 - Enviando/recebendo partial e full-routing

- Enviando parcial para o meu DOWNSTREAM

- A lógica é mesma, inclusive posso aproveitar a mesma as-path acl "2"

G**X**-R1(config-router)#neighbor 187.16.216.99 filter-list 2 out

!--- vamos aplicar o mesmo filter-list de entrada de UPSTREAMS para o DOWNSTREAM

OBS: FILTER LIST = AS-PATH ACCESS-LIST

4.7
ROUTE-MAP x PREFIX-LIST

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **PREFIX-LIST**:
  - São muito simples de entender e configurar, porém impossibilitam qualquer modificação, apenas ”aceitam” e ”descartam” prefixos
  - Cada regra de ”permit” e ”deny” de fato ”aceitam” e ”descartam” o prefixo correspondente

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - Proporcionam uma robustez muito maior nos filtros, porém uma complexidade diretamente proporcional de configuração
  - Possibilitam alterações em certos campos nas rotas como por exemplo: next-hop, as-path, route tag, metric, local preference, weight, community, e muitos outros não só em BGP

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - É composto de 5 campos:
    1. Nome { string }
    2. Ação { permit | deny }
    3. Número de seq { 0 – 65535 }
    4. Condições
    5. Modificações (opcional)

```
router(config)#route-map FILTROS-DE-ENTRADA permit 10
router(config-route-map)#match ip address prefix-list MUDAR-METRIC
router(config-route-map)#set metric 50
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **NOME: FILTROS-DE-ENTRADA**
  - Campo tipo string: aceita letras ”–” e ”_”
  - É declarado na configuração do neighbor BGP
  - É case sensitive

```
router(config)#route-map FILTROS-DE-ENTRADA permit 10
router(config-route-map)#match ip address prefix-list MUDAR-METRIC
router(config-route-map)#set metric 50
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **AÇÃO: { permit | deny }**
  - Define a ação a ser tomada caso 100% das condições dentro do route-map sejam atendidas

```
router(config)#route-map FILTROS-DE-ENTRADA permit 10
router(config-route-map)#match ip address prefix-list MUDAR-METRIC
router(config-route-map)#set metric 50
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **Seq number: { 0 … 65535 }**
  - A ordem que esse route-map vai ser processado na hierarquia, seguindo a mesma lógica dos ”seq” da prefix-list

```
router(config)#route-map FILTROS-DE-ENTRADA permit 10
router(config-route-map)#match ip address prefix-list MUDAR-METRIC
router(config-route-map)#set metric 50
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **Condições: match ip address prefix-list MUDAR-METRIC**
  - Para receber a ação ”permit” e as modificações opcionais (set metric 50) o prefixo precisaria estar contido na prefix list ”**MUDAR-METRIC**”
  - OBS: é possível estabelecer mais de uma condição, por exemplo:
    match ip address prefix-list MUDAR-METRIC
    match as-path 10
    match community 60
  - **Mas nesse caso é preciso que todas as condições sejam atendidas, caso contrário, o route-map é ignorado e é processado o próximo route-map na sequência**

```
router(config)#route-map FILTROS-DE-ENTRADA permit 10
router(config-route-map)#match ip address prefix-list MUDAR-METRIC
router(config-route-map)#set metric 50
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **Modificações: set metric 50**
  - Ao atender 100% das condições podem ser aplicadas modificações opcionais nas rotas
  - OBS: também é possível realizar mais de uma modificação, por exemplo:
    set local-preference 120
    set weight 40
    set community 123:321

```
router(config)#route-map FILTROS-DE-ENTRADA permit 10
router(config-route-map)#match ip address prefix-list MUDAR-METRIC
router(config-route-map)#set metric 50
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **Route-map sem condições declaradas:**
  - Nesses casos TODAS as rotas receberão à ação {permit | deny} e qualquer modificação opcional, que nesse exemplo foi alterar o local preference para 200

```
router(config)#route-map FILTROS-DE-ENTRADA permit 10
router(config-route-map)#set local-preference 200
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **Route-map sem condições declaradas:**
  - Caso hajam outros route-maps abaixo (números maiores de seq) de um route-map sem condições declaradas, esses jamais serão processados e serão totalmente inúteis

```
router#sh running-config | begin TESTE
route-map TESTE permit 1
 set weight 10
!
route-map TESTE permit 2
 match ip address BLOCOS-CLIENTE-1
 set metric 10
 set local-preference 350
!
route-map TESTE permit 3
 match as-path 20
 set local-preference 400
!
```

4.7 - Route-map x prefix-list

- Políticas de filtro baseadas em **ROUTE-MAP**:
  - **Quando um "PERMIT" se transforma em "DENY"**
  - Ao combinar route-map e prefix-lists, entedam o "permit" da prefix list como um "**faz parte da lista**" e não como um "**aceitar**" propriamente dito, pois a ação de "**aceitar**" vai depender do que for declarado na "ação" do route-map
  - Se é um route-map com ação DENY, tudo que tiver como **permit** na prefix list será **bloqueado,** por exemplo:

```
!
ip prefix-list A seq 5 permit 192.168.2.0/24
ip prefix-list A seq 10 deny 192.168.44.0/24
!
route-map TESTE deny 1
 match ip address prefix-list A
!
route-map TESTE permit 2
!
```

  - Nesse caso o prefixo 192.168.2.0/24 será **bloqueado**, enquanto 192.168.44.0/24 será **permitido**

4.8
AS-PATH PREPEND

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.8 - AS-Path Prepend

- Para manipular o download pra os prefixos do meu próprio AS eu posso facilmente anunciar prefixos mais ou menos específicos

- Mas como AS de trânsito não tenho como pedir aos meus clientes que modifiquem seus anúncios caso eu queira usar um link específico

- O AS PATH prepend adiciona números extras de AS no AS PATH das rotas anunciadas para um determinado neighbor

4.9
MANIPULANDO TRÂNSITO DOS DOWNSTREAMS COM PREPEND

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

4.9 - Manipulando trânsito dos downstreams com prepend

- LAB: Todos os grupos configurar Upstreams + dowmstreams de acordo com os sequintes requisitos:

  - *Apenas o download do downstream deverá chegar pelo AS 100, de forma que o tráfego inbound do AS local continuará chegando pelo AS 200*

4.9 - Manipulando trânsito dos downstreams com prepend

- Cofigurando:

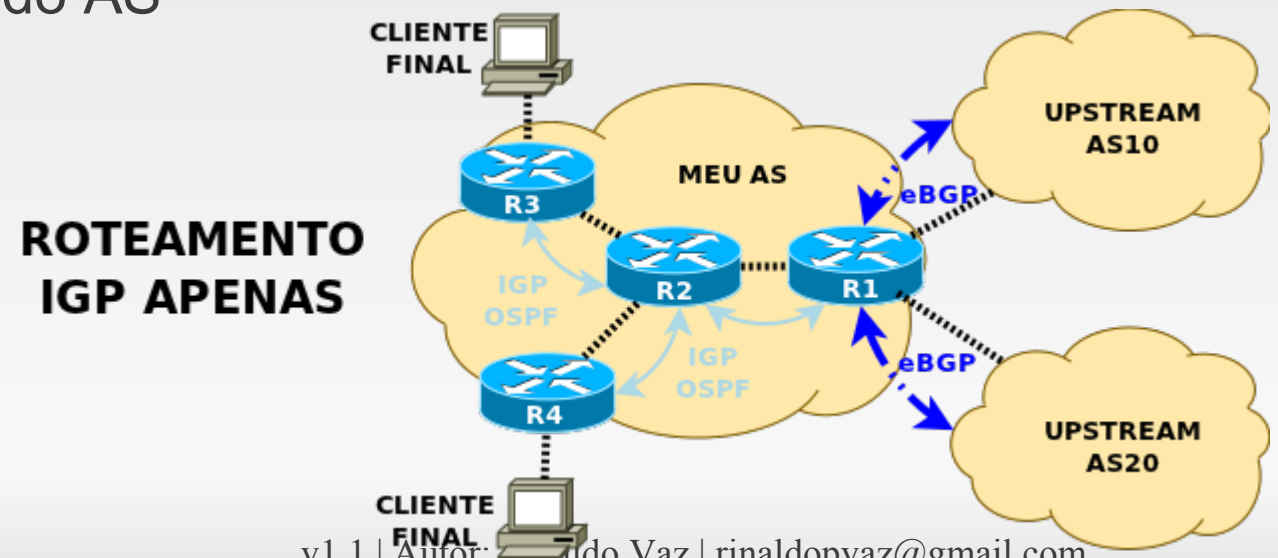
G**X**-R1(config-router)#no neighbor 200.**X**.1.1 prefix-list ANUNCIAR-AS-200 out
!--- desconfigura a prefix list ANUNCIAR-AS-200 do filtro de saida do AS 200
G**X**-R1(config-router)#neighbor 200.**X**.1.1 route-map **AS-200-OUT** out
!--- configura o route-map AS-200-OUT como filtro de saída para o AS 200
G**X**-R1(config-router)#exit
G**X**-R1(config)#route-map **AS-200-OUT** permit 1
!--- entra no modo de configuração do route-map seq 1
G**X**-R1(config-route-map)#match ip address prefix-list BLOCOS-AS-99
!--- declara a condição para receber "permit" + a modificação opcional prepend 4 vezes
G**X**-R1(config-route-map)#set as-path prepend **X X X X**
!--- insere mais 4 vezes o AS local no AS PATH
*G**X**-R1(config)#route-map **AS-200-OUT** permit 2*
!--- entra no modo de configuração do route-map seq 2
G**X**-R1(config-route-map)#match ip address prefix-list MEUS-BLOCOS
!--- declara a condição para receber "permit", não haverão modificaçõs de prepende para
!--- os blocos do AS. Caso ainda não tenha sido criada, é preciso criar a prefix-list MEUS-BLOCOS
!--- e adicionar os prefixo do AS dentro dela

5.1
IMPLEMENTANDO MÚLTIPLAS SAÍDAS PARA CLIENTES DE TRÂNSITO

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

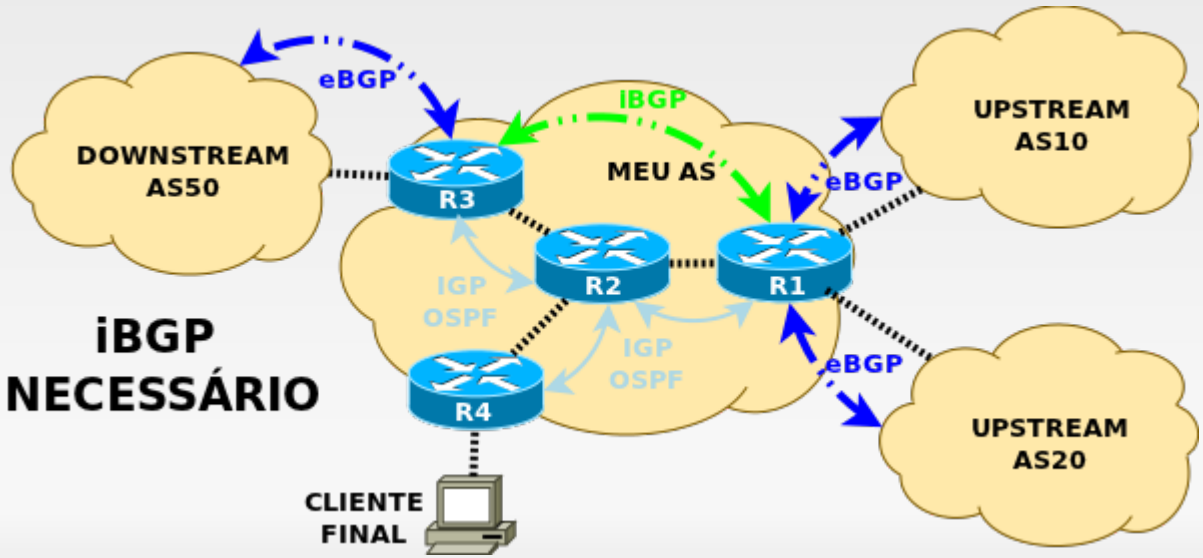
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Conforme discutido anteriormente, o BGP é um protocolo desenvolvido para troca de rotas entre AS's distintos.
- Deve ser evitado dentro do AS, a menos que as extremidades precisem trocar rotas de AS's externos.
- Em outras palavras, se houver realmente a necessidade de entregar para um DOWNSTREAM rotas recebidas de um UPSTREAM
- No exemplo abaixo não há necessidade de injetar rotas externas dentro do AS



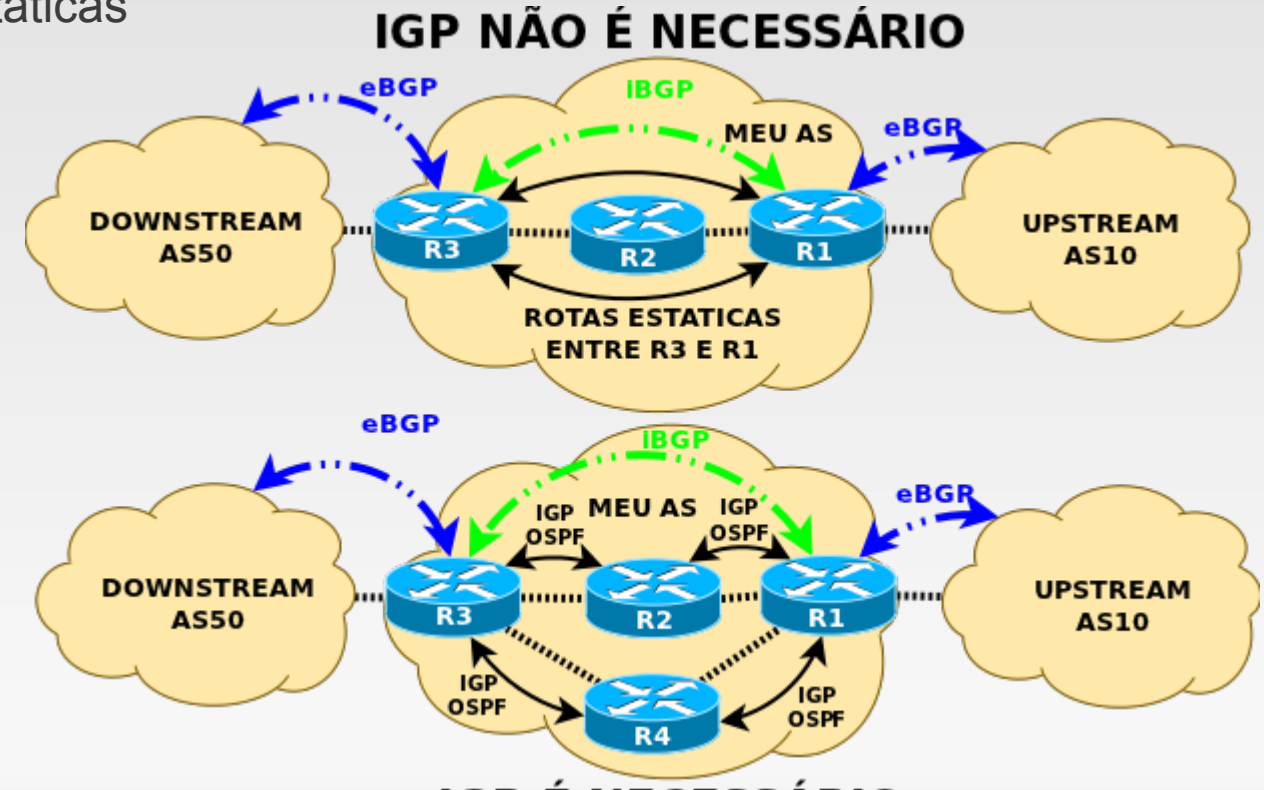
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Injetar rotas externas se faz necessário apenas quando essas precisam ”atravessar” o AS
- Em outras palavras, se houver realmente a necessidade de entregar para um DOWNSTREAM rotas recebidas de um UPSTREAM
- No exemplo as rotas recebidas do AS 20 e 30 precisam chegar ao AS 50 e vice-versa



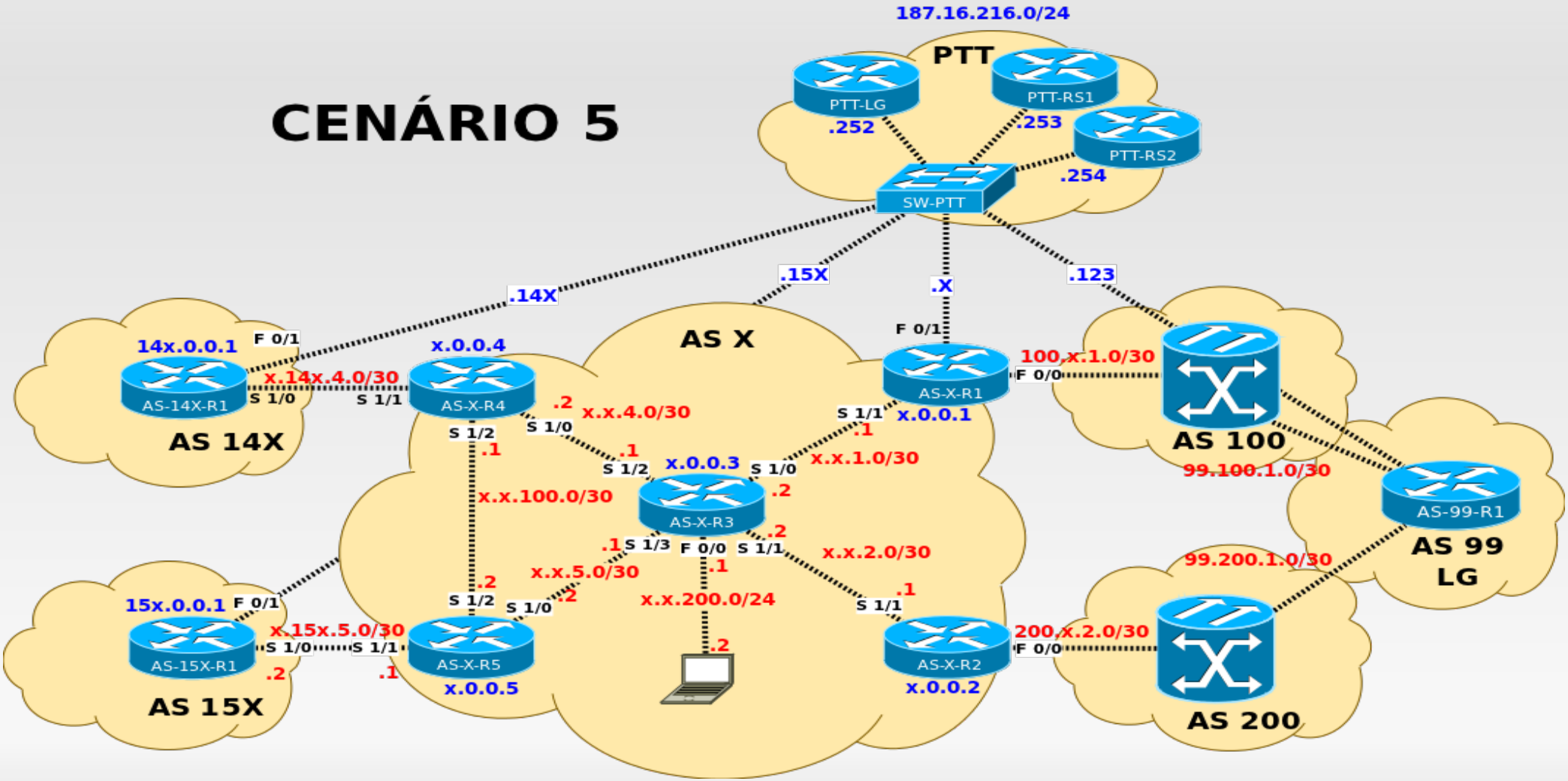
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Alguns pontos importantes:
  1. Um protocolo de roteamento IGP precisa garantir conectividade IP entre TODOS os routers do AS
  2. Mas se não houver múltiplos caminhos, não há necessidade de IGP, apenas rotas estáticas



5.1 - Implementando múltiplas saídas para clientes de trânsito

- Em nosso LAB há múltiplos caminhos, e todos os routers (exceto AS's externos) já estão configurados com OSPF



5.1 - Implementando múltiplas saídas para clientes de trânsito

- Nos próximos LABs os alunos se juntarão em grupos de 3
- Alunos da mesma bancada farão parte do mesmo grupo, de maneira que os grupos 1,2,3 serão o **GRUPO 1 – AS 1**
- Grupos 4, 5 e 6 serão **GRUPO 2 – AS 2**
- Grupos 7, 8 e 9 serão **GRUPO 3 – AS 3**
- Grupos 10, 11 e 12 serão **GRUPO 4 – AS 4**
- Grupos 13, 14 e 15 serão **GRUPO 5 – AS 5**
- Grupos 16, 17 e 18 serão **GRUPO 6 – AS 6**
- Grupos 19, 20 e 21 serão **GRUPO 7 – AS 7**
- Grupos 22, 23 e 24 serão **GRUPO 8 – AS 8**

5.1 - Implementando múltiplas saídas para clientes de trânsito

- O router R3 (router central do AS) responde pelo IP **X**.**X**.100.1/24

- Cada membro do grupo vai configurar um IP da mesma rede em seu laptop, exemplo sugerido laptop1 **X.X.100.2/24**, laptop2 **X.X.100.3/24** e laptop3 **X.X.100.4/24**
- Testar conectividade IP com R3 e a partir dele acessar todos os outros routers via telnet

5.1 - Implementando múltiplas saídas para clientes de trânsito

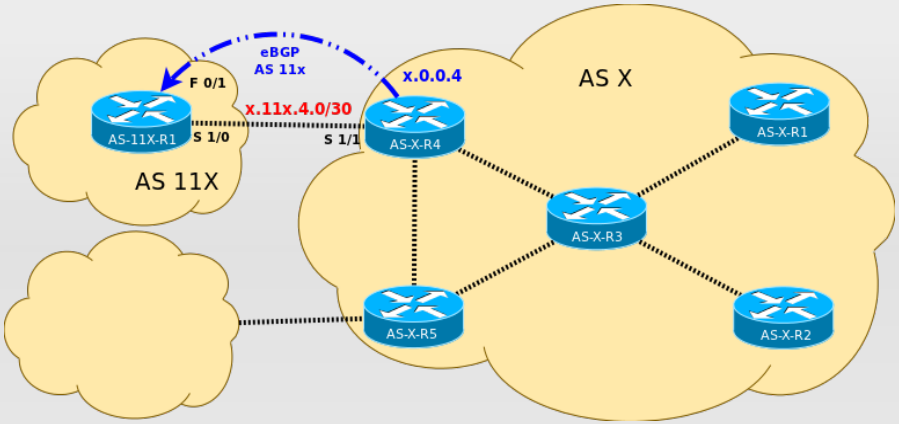
- LAB: Considerando que as cidades onde ficam R4 e R5 atendem clientes de trânsito (DOWNSTREAMS), e as cidades onde ficam R1 e R2 tem saída de trânsito IP (UPSTREAMS), precisaremos que R4 e R5 recebam e enviem rotas para o AS 100 (através de R1), e para o AS 200 através de R2

- Vamos estabelecer sessões eBGP entre os clientes de trânsito AS 11x com R4 e AS 12x com R5.

- NOTA:

!--- No intuito de facilitar a visualização das rotas por parte dos alunos, em
!--- todos os LABS seguintes não trabalharemos com anúncios mais
!--- específicos que /8, (com excessão dos anúncios ao PTT no lab
!--- correspondente

5.1 - Implementando múltiplas saídas para clientes de trânsito



- Configurando eBGP com o downstream:

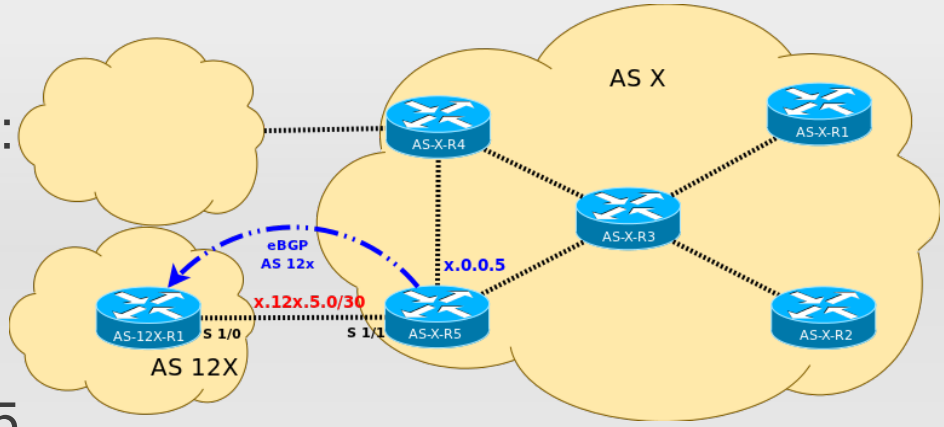
AS 14**X** em R4:

AS-**X**-R4#conf t
AS-**X**-R4(config)#router bgp **X**
AS-**X**-R4(config-router)#bgp router-id **X**.0.0.4
!--- nesse LAB vamos configurar um Router-ID para facilitar a visualização das rotas
AS-**X**-R4(config-router)#neighbor **X**.14**X**.4.2 remote-as 14**X**
AS-**X**-R4(config-router)#neighbor **X**.14**X**.4.2 description eBGP-downstream-AS-14**X**
AS-**X**-R4(config-router)#neighbor **X**.14**X**.4.2 soft-reconfiguration inbound
AS-**X**-R4(config-router)#neighbor **X**.14**X**.4.2 prefix-list BLOCOS-AS-14**X** in
!--- declara o filtro de entrada para rotas recebidas desse DOWNSTREAM
AS-**X**-R4(config-router)#exit
AS-**X**-R4(config)#ip prefix-list BLOCOS-AS-14**X** permit 14**X**.0.0.0/8
!--- adiciona os blocos CIDR do DOWNSTREAM nesa prefix-list que foi
!--- declarada nas configurações BGP

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Configurando eBGP com o downstream:

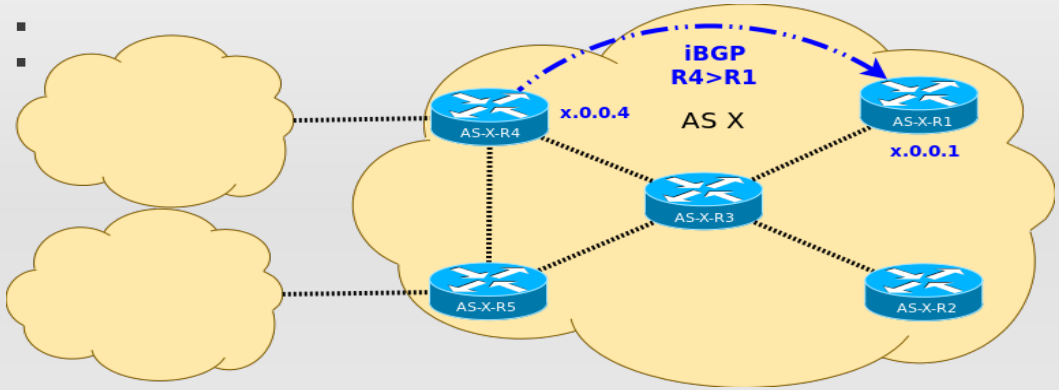
AS 15**X** em R5:



AS-**X**-R5#conf t
AS-**X**-R5(config)#router bgp **X**
AS-**X**-R5(config-router)#bgp router-id **X**.0.0.5
!--- nesse LAB vamos configurar um Router-ID para facilitar a visualização das rotas
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 remote-as 15**X**
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 description eBGP-downstream-AS-15**X**
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 soft-reconfiguration inbound
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 prefix-list BLOCOS-AS-15**X** in
!--- declara o filtro de entrada para rotas recebidas desse DOWNSTREAM
AS-**X**-R5(config-router)#exit
AS-**X**-R5(config)#ip prefix-list BLOCOS-AS-15**X** permit 15**X**.0.0.0/8
!--- adiciona os blocos CIDR do DOWNSTREAM nesa prefix-list que foi
!--- declarada nas configurações BGP

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Configurando iBGP entre R4 e R1:

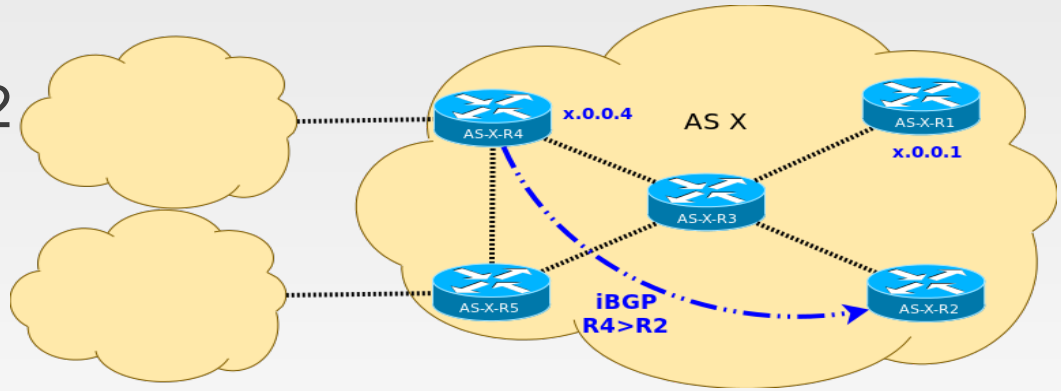


AS-X-R4#conf t

AS-X-R4(config)#router bgp X

AS-X-R4(config-router)#neighbor X.0.0.1 remote-as X
AS-X-R4(config-router)#neighbor X.0.0.1 description iBGP Router 1

- Configurando iBGP entre R4 e R2



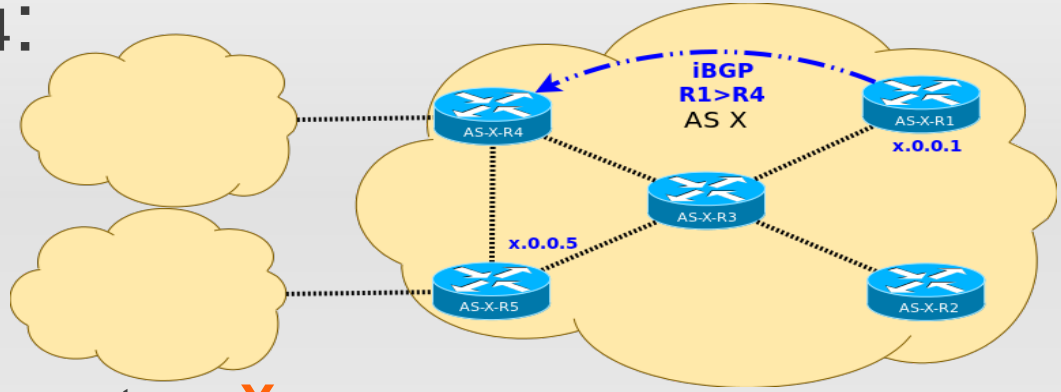
AS-X-R4#conf t

AS-X-R4(config)#router bgp X

AS-X-R4(config-router)#neighbor X.0.0.2 remote-as X
AS-X-R4(config-router)#neighbor X.0.0.2 description iBGP Router 2

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Configurando iBGP entre R1 e R4:

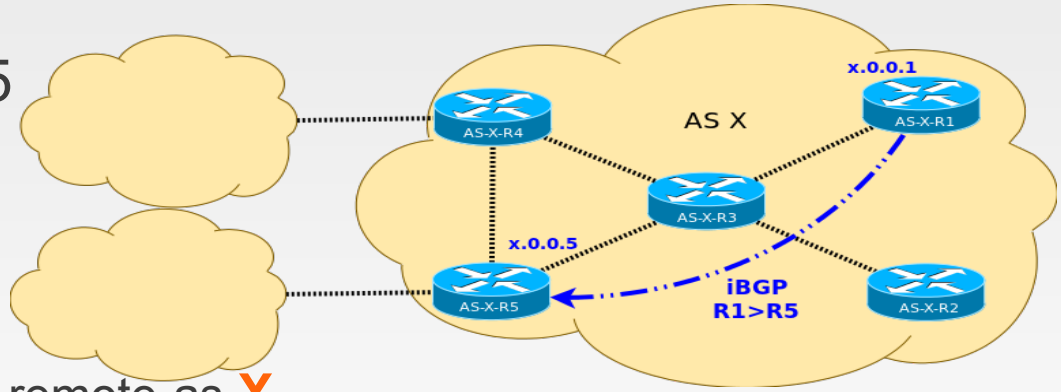


AS-X-R1#conf t

AS-X-R1(config)#router bgp X

AS-X-R1(config-router)#neighbor X.0.0.4 remote-as X
AS-X-R1(config-router)#neighbor X.0.0.4 description iBGP Router 4

- Configurando iBGP entre R1 e R5



AS-X-R4#conf t

AS-X-R4(config)#router bgp X

AS-X-R4(config-router)#neighbor X.0.0.5 remote-as X
AS-X-R4(config-router)#neighbor X.0.0.5 description iBGP Router 5

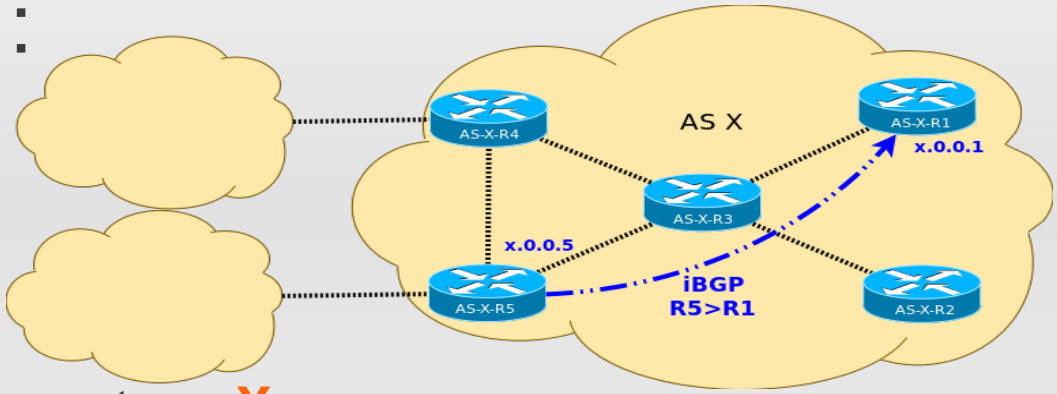
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Configurando iBGP entre R5 e R1:

AS-X-R5#conf t

AS-X-R5(config)#router bgp X

AS-X-R5(config-router)#neighbor X.0.0.1 remote-as X
AS-X-R5(config-router)#neighbor X.0.0.1 description iBGP Router 1

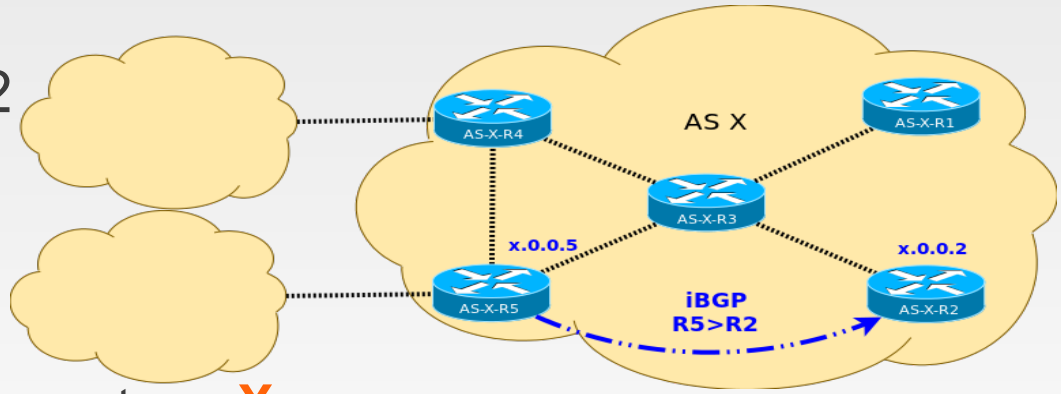


- Configurando iBGP entre R5 e R2

AS-X-R5#conf t

AS-X-R5(config)#router bgp X

AS-X-R5(config-router)#neighbor X.0.0.2 remote-as X
AS-X-R5(config-router)#neighbor X.0.0.2 description iBGP Router 2



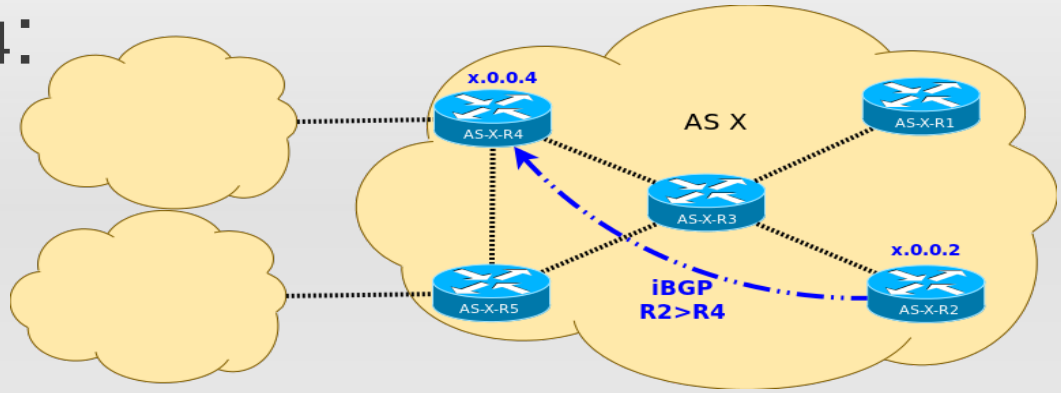
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Configurando iBGP entre R2 e R4:

AS-X-R2#conf t

AS-X-R2(config)#router bgp X

AS-X-R2(config-router)#neighbor X.0.0.4 remote-as X
AS-X-R2(config-router)#neighbor X.0.0.4 description iBGP Router 4

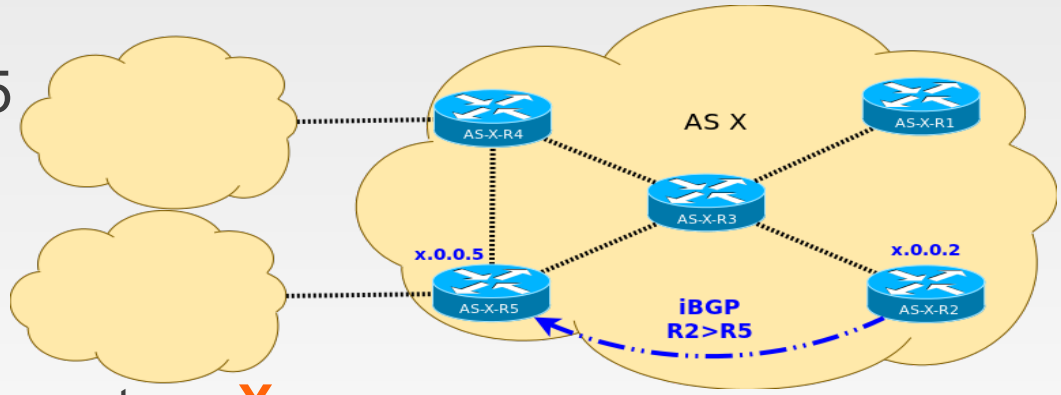


- Configurando iBGP entre R2 e R5

AS-X-R2#conf t

AS-X-R2(config)#router bgp X

AS-X-R2(config-router)#neighbor X.0.0.5 remote-as X
AS-X-R2(config-router)#neighbor X.0.0.5 description iBGP Router 5



5.1 - Implementando múltiplas saídas para clientes de trânsito

- As sessões BGP subiram? Provavelmente não!
- Como decidimos estabelecer as sessões utilizando os IPs de loopback dos routers internos, nos deparamos com um problema de conexão TCP
- A menos que especifiquemos o contrário, o IP de origem de todo pacote é o IP da interface que saiu
- Em outras palavras, se a melhor rota para alcançar R4 é pela serial 1/1, o IP de origem será o IP dessa interface

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Vejamos: 

AS-X-R4#sh ip route X.0.0.1
Routing entry for X.0.0.1/32
Known via "ospf 1", distance 110, metric 129, type intra area
Last update from X.X.4.1 on Serial1/0, 07:43:20 ago
Routing Descriptor Blocks:
** X.X.4.1, from X.0.0.1, 07:43:20 ago, via Serial1/0*
Route metric is 129, traffic share count is 1

- Nesse caso, quando o pacote "syn" chegar ao router R1 (X.0.0.1) para solicitar o início da sessão BGP, o seu IP de origem desse pacote será X.X.4.2, e já que R1 foi configurado para estabelecer BGP com X.0.0.4 (R4) ele naturalmente rejeita essa conexão
- Analizando o debug:

07:42:42: BGP: X.0.0.1 open active, delay 7672ms
07:42:50: BGP: X.0.0.1 open active, local address X.X.4.2
07:42:50: BGP: X.0.0.1 open failed: Connection refused by remote host

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Para resolver esse problema precisamos dizer ao meu router R4 que utilize o IP ***X***.0.0.4 para estabelecer a conexão TCP-179 com R1


```
AS-X-R4(config)#router bgp X
AS-X-R4(config-router)#neighbor X.0.0.1 update-source loopback 0
AS-X-R4(config-router)#neighbor X.0.0.2 update-source loopback 0
!--- loopback 0 é a interface que possui o IP X.0.0.4 no R4
!--- mas também podemos declarar um IP ao invés de uma interface
```

- Analizando novamente o debug:

```
07:48:50: BGP: X.0.0.1 open active, local address X.0.0.4
07:48:50: BGP: X.0.0.1 went from Active to OpenSent
07:48:50: BGP: X.0.0.1 sending OPEN, version 4, my as: X
07:48:50: BGP: X.0.0.1 rcv message type 1, length (excl. header) 26
07:48:50: BGP: X.0.0.1 rcv OPEN, version 4
07:48:50: BGP: X.0.0.1 rcv OPEN w/ OPTION parameter len: 16
07:48:50: BGP: X.0.0.1 rcvd OPEN w/ optional parameter type 2 (Capability) len 6
07:48:50: BGP: X.0.0.1 OPEN has CAPABILITY code: 1, length 4
07:48:50: BGP: X.0.0.1 OPEN has MP_EXT CAP for afi/safi: 1/1
07:48:50: BGP: X.0.0.1 rcvd OPEN w/ optional parameter type 2 (Capability) len 2
07:48:50: BGP: X.0.0.1 OPEN has CAPABILITY code: 128, length 0
07:48:50: BGP: X.0.0.1 OPEN has ROUTE-REFRESH capability(old) for all address-families
07:48:50: BGP: X.0.0.1 rcvd OPEN w/ optional parameter type 2 (Capability) len 2
07:48:50: BGP: X.0.0.1 OPEN has CAPABILITY code: 2, length 0
07:48:50: BGP: X.0.0.1 OPEN has ROUTE-REFRESH capability(new) for all address-families
07:48:50: BGP: X.0.0.1 went from OpenSent to OpenConfirm
07:48:50: BGP: X.0.0.1 went from OpenConfirm to Established
07:48:50: %BGP-5-ADJCHANGE: neighbor X.0.0.1 Up
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- … fazer a mesma mudança em R5


```
AS-X-R5(config)#router bgp X
AS-X-R5(config-router)#neighbor X.0.0.1 update-source loopback 0
AS-X-R5(config-router)#neighbor X.0.0.2 update-source loopback 0
!--- loopback 0 é a interface que possui o IP X.0.0.5 no R5
```

07:48:50: %BGP-5-ADJCHANGE: neighbor X.0.0.1 Up

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Por que não foi necessário realizar a mesma configuração em R1?

  *Por que o R1 responde (com a mensagem type 1) utilizando o IP destino que R4 inseriu no pacote de "OPEN" (**X**.0.0.1)*

- Na prática é preciso que apenas um dos dois routers altere o "update-source", porém é recomendável alterar nos dois, pois isso pode fazer com que a sessão estabeleca alguns segundos mais rápido numa eventual queda.

- configurando o update-source nas sessões R1<->R4 e R1<->R5

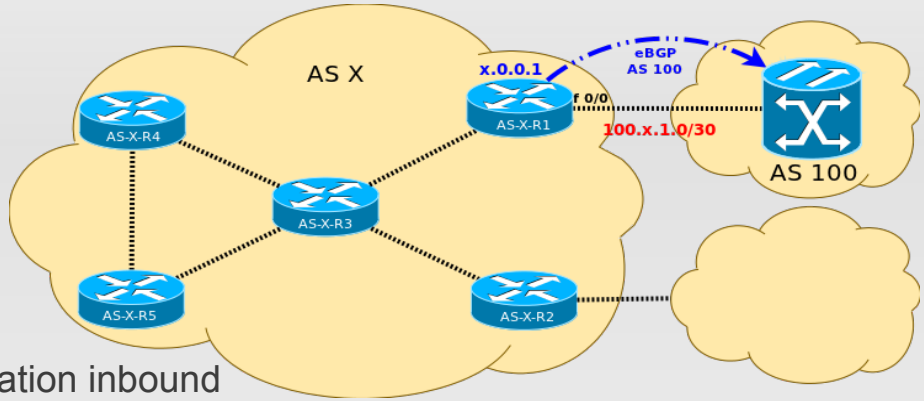

```
AS-X-R1#conf t
AS-X-R1(config)#router bgp X
AS-X-R1(config-router)#neighbor X.0.0.4 update-source loopback 0
AS-X-R1(config-router)#neighbor X.0.0.5 update-source loopback 0
```

- configurando o update-source nas sessões R2<->R4 e R2->R5


```
AS-X-R2#conf t
AS-X-R2(config)#router bgp X
AS-X-R2(config-router)#neighbor X.0.0.4 update-source loopback 0
AS-X-R2(config-router)#neighbor X.0.0.5 update-source loopback 0
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

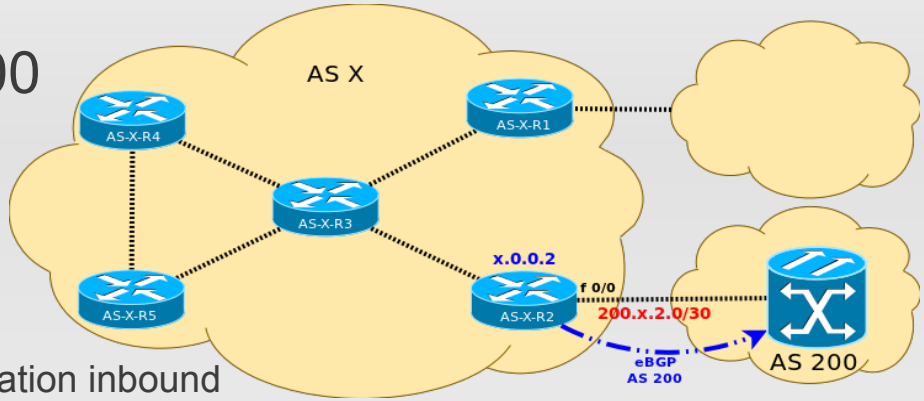
- Vamos também configurar as sessões BGP com os UPSTREAMS
  - Primeiro eBGP entre R1 e AS 100




```
AS-X-R1#conf t
AS-X-R1(config)#router bgp X
AS-X-R1(config-router)#network X.0.0.0
AS-X-R1(config-router)#neighbor 100.X.1.1 remote-as 100
AS-X-R1(config-router)#neighbor 100.X.1.1 soft-reconfiguration inbound
AS-X-R1(config-router)#neighbor 100.X.1.1 route-map ANUNCIAR-AS-100 out
AS-X-R1(config-router)#exit
AS-X-R1(config)#ip prefix-list MEUS-BLOCOS permit X.0.0.0/8
AS-X-R1(config)#ip prefix-list BLOCOS-AS-14X permit 14X.0.0.0/8
AS-X-R1(config)#ip prefix-list BLOCOS-AS-15X permit 15X.0.0.0/8

AS-X-R1(config)#route-map ANUNCIAR-AS-100 permit 10
AS-X-R1(config-route-map)#match ip address prefix-list MEUS-BLOCOS
AS-X-R1(config-route-map)#exit
AS-X-R1(config)#route-map ANUNCIAR-AS-100 permit 20
AS-X-R1(config-route-map)#match ip address prefix-list BLOCOS-AS-14X
AS-X-R1(config-route-map)#exit
AS-X-R1(config)#route-map ANUNCIAR-AS-100 permit 30
AS-X-R1(config-route-map)#match ip address prefix-list BLOCOS-AS-15X
AS-X-R1(config-route-map)#end
AS-X-R1#clear ip bgp * soft
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Vamos também configurar as sessões BGP com os UPSTREAMS
  - Em seguida eBGP entre R2 e AS 200




```
AS-X-R2#conf t
AS-X-R2(config)#router bgp X
AS-X-R2(config-router)#network X.0.0.0
AS-X-R2(config-router)#neighbor 200.X.2.1 remote-as 200
AS-X-R2(config-router)#neighbor 200.X.2.1 soft-reconfiguration inbound
AS-X-R2(config-router)#neighbor 200.X.2.1 route-map ANUNCIAR-AS-200 out
AS-X-R2(config-router)#exit
AS-X-R2(config)#ip prefix-list MEUS-BLOCOS permit X.0.0.0/8
AS-X-R2(config)#ip prefix-list BLOCOS-AS-14X permit 14X.0.0.0/8
AS-X-R2(config)#ip prefix-list BLOCOS-AS-15X permit 15X.0.0.0/8

AS-X-R2(config)#route-map ANUNCIAR-AS-200 permit 10
AS-X-R2(config-route-map)#match ip address prefix-list MEUS-BLOCOS
AS-X-R2(config-route-map)#exit
AS-X-R2(config)#route-map ANUNCIAR-AS-200 permit 20
AS-X-R2(config-route-map)#match ip address prefix-list BLOCOS-AS-14X
AS-X-R2(config-route-map)#exit
AS-X-R2(config)#route-map ANUNCIAR-AS-200 permit 30
AS-X-R2(config-route-map)#match ip address prefix-list BLOCOS-AS-15X
AS-X-R2(config-route-map)#end
AS-X-R2#clear ip bgp * soft
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Estabelecidas as sessões iBGP, vamos aos "novos" problemas:

AS-X-R1#sh ip bgp neighbors 100.X.1.1 advertised-routes

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> X.0.0.0          0.0.0.0                  0         32768 i
```

AS-X-R2#sh ip bgp neighbors 200.X.2.1 advertised-routes

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> X.0.0.0          0.0.0.0                  0         32768 i
```

- Notem que não estamos anunciando as redes recebidas pelos meus DOWNSTREAMS, mas apenas o classe A do AS local

- Por que?

5.1 - Implementando múltiplas saídas para clientes de trânsito

- O primeiro critério para o protocolo BGP anunciar algum prefixo é que esse prefixo esteja válido na FIB

AS-X-R1#sh ip bgp

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> X.0.0.0          0.0.0.0                  0         32768 i
* i14X.0.0.0        6.116.4.2                0    100          0 14X i
* i15X.0.0.0        6.126.5.2                0    100          0 15X i
...
```

- O prefixo "X.0.0.0" está com o caracter ">", o que demonstra que esse prefixo está de fato na FIB (best)
- Porém não é o que acontece com os prefixos recebidos dos DOWNSTREAMS
- Os prefixos 14X.0.0.0 e 15X.0.0.0 estão na RIB, mas não na FIB
- Por que esses prefixos não estão na FIB?

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Vamos verificar detalhadamente o status de um desses prefixos em R1...

AS-X-R1#sh ip bgp 14X.0.0.0
BGP routing table entry for 14X.0.0.0/8, version 0
Paths: (1 available, **no best path**)
  Not advertised to any peer
  14X
    X.14X.4.2 (**inaccessible**) from X.0.0.4 (X.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal, **not synchronized**

- Um problema é perceptível sem muito trabalho:

  "X.14X.4.2 (inaccessible)"

- Para que esse prefixo seja válido na FIB é necessário que o "next-hop" correspondente também seja válido, o que não ocorre
- Mas esse é apenas um dos problemas perceptíveis na saída do último comando

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Resolvendo o primeiro problema:
- Não é recomendável que as interfaces externas do AS participem do IGP (ospf), portanto vamos para outra solução
- Vamos então utilizar o comando BGP "next-hop-self" que altera o next-hop para o IP de neighbor


```
AS-X-R1#conf t
AS-X-R1(config)#router bgp X
AS-X-R1(config-router)#neighbor X.0.0.4 next-hop-self
AS-X-R1(config-router)#neighbor X.0.0.5 next-hop-self
```


```
AS-X-R2#conf t
AS-X-R2(config)#router bgp X
AS-X-R2(config-router)#neighbor X.0.0.4 next-hop-self
AS-X-R2(config-router)#neighbor X.0.0.5 next-hop-self
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Continuando em R4 e R5...


```
AS-X-R4#conf t
AS-X-R4(config)#router bgp X
AS-X-R4(config-router)#neighbor X.0.0.1 next-hop-self
AS-X-R4(config-router)#neighbor X.0.0.2 next-hop-self
```


```
AS-X-R5#conf t
AS-X-R5(config)#router bgp X
AS-X-R5(config-router)#neighbor X.0.0.1 next-hop-self
AS-X-R5(config-router)#neighbor X.0.0.2 next-hop-self
```

- Vamos agora verificar as mudanças...

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Vamos detalher novamente um dos prefixos em R1:

AS-X-R1#sh ip bgp 14X.0.0.0
BGP routing table entry for 14X.0.0.0/8, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  14X
    X.0.0.5 (metric 129) from X.0.0.5 (X.0.0.5)
      Origin IGP, metric 0, localpref 100, valid, internal, **not synchronized**

- Notem que o next-hop não se encontra mais inacessível, porém a rota mostra uma outra mensagem: "**NOT SYNCRONIZED**"
- Isso acontece com rotas aprendidas por iBGP, uma vez que por padrão o protocolo exige a sincronização da rota antes de anuncia-la aos AS's externos
- Nas versões e caixas mais novas o ”syncronize” vem desabilitado

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Resolvendo o problema de maneira "duvidosa" em R1 e R2...

AS-**X**-R1#conf t
AS-**X**-R1(config)#router bgp **X**
AS-**X**-R1(config-router)#no synchronization
AS-**X**-R1(config-router)#end
AS-**X**-R1#clear ip bgp *

AS-**X**-R2#conf t
AS-**X**-R2(config)#router bgp **X**
AS-**X**-R2(config-router)#no synchronization
AS-**X**-R2(config-router)#end
AS-**X**-R2#clear ip bgp *

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Continuando em R4 e R5...

AS-**X**-R4#conf t
AS-**X**-R4(config)#router bgp **X**
AS-**X**-R4(config-router)#no synchronization
AS-**X**-R4(config-router)#end
AS-**X**-R4#clear ip bgp *

AS-**X**-R5#conf t
AS-**X**-R5(config)#router bgp **X**
AS-**X**-R5(config-router)#no synchronization
AS-**X**-R1(config-router)#end
AS-**X**-R5#clear ip bgp *

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Conferindo as mudanças em R1 e demais

AS-X-R1#sh ip bgp

```
...
   Network          Next Hop            Metric LocPrf Weight Path
*> X.0.0.0          0.0.0.0                  0         32768 i
*> 100.0.0.0        100.X.1.1                0             0 100 i
*>i14X.0.0.0        X.0.0.4                  0    100      0 146 i
*>i15X.0.0.0        X.0.0.5                  0    100      0 156 i
```

- Verificando o caracter ">" podemos ter certeza dessa vez as rotas foram para FIB

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Vamos agora ter certeza que os prefixos do meus DOWNSTREAMS estão sendo anunciados aos UPSTREAMS

AS-X-R1#sh ip bgp 14X.0.0.0
BGP routing table entry for 14X.0.0.0/8, version 16
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  100.X.1.1
  11X
    X.0.0.4 (metric 129) from X.0.0.4 (X.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal, best

- Bingo!! Estamos anunciando os prefixos de DOWNSTREAMS para o AS 100
- O mesmo vale para R2, R4 e R5, é só conferir

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Agora entrem no **Looking Glass (**nesse LAB o IP mudou para **99.99.99.1**) e testem a conectividade IP para os AS's clientes 14**X** e 14**X**,

```
telnet 99.99.99.1...

!--executem traceroute para os IPs dos seus respectivos downstreams

AS-99-R1>traceroute 14X.0.0.1

Type escape sequence to abort.
Tracing the route to 14X.0.0.1

  1 99.100.1.2 12 msec 16 msec 4 msec
  2 100.X.1.2 [AS 100] 60 msec 28 msec 36 msec
  3  *  *  *
  4  *  *  *
  5  *  *  *
```

- **CONCLUSÕES:** Se o "syncronization" existe, precisamos compreender sua função antes de desabilita-lo

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Entendendo o "syncronization"...
- Uma proteção natural que o BGP implementa é exigir a sincronização de uma rota aprendida por iBGP antes de inseri-la na FIB
- Quando um router recebe um prefixo de origem "internal" através de iBGP esse não tem como saber se os routers intermediários também conhecem rota para esse prefixo
- Vamos entender melhor...

5.1 - Implementando múltiplas saídas para clientes de trânsito

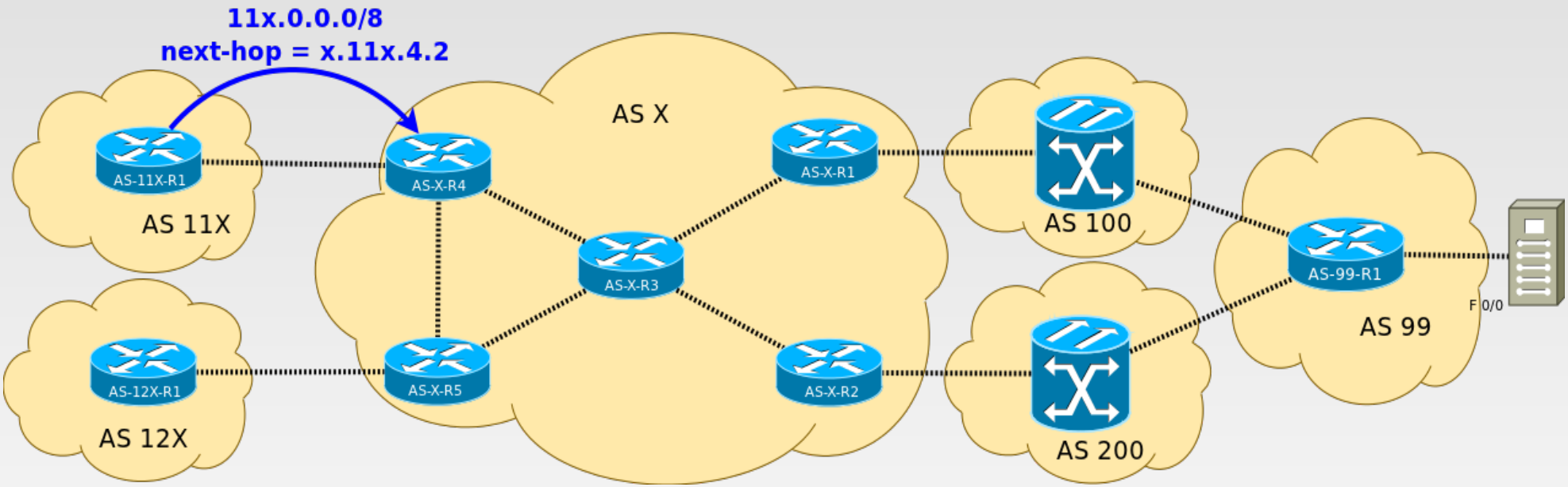
- Entendendo o ”syncronization”...

- R4 estabelece um iBGP diretamente com R1 e anuncia os prefixos recebidos do AS 14**X**. O problema é que AS-**X**-R1 não tem como saber se R3 conhece rota para os prefixos do AS 14**X**

- O syncronization é um mecanismo de proteção que evita loops de roteamento, vamos entender melhor o que aconteceu no LAB

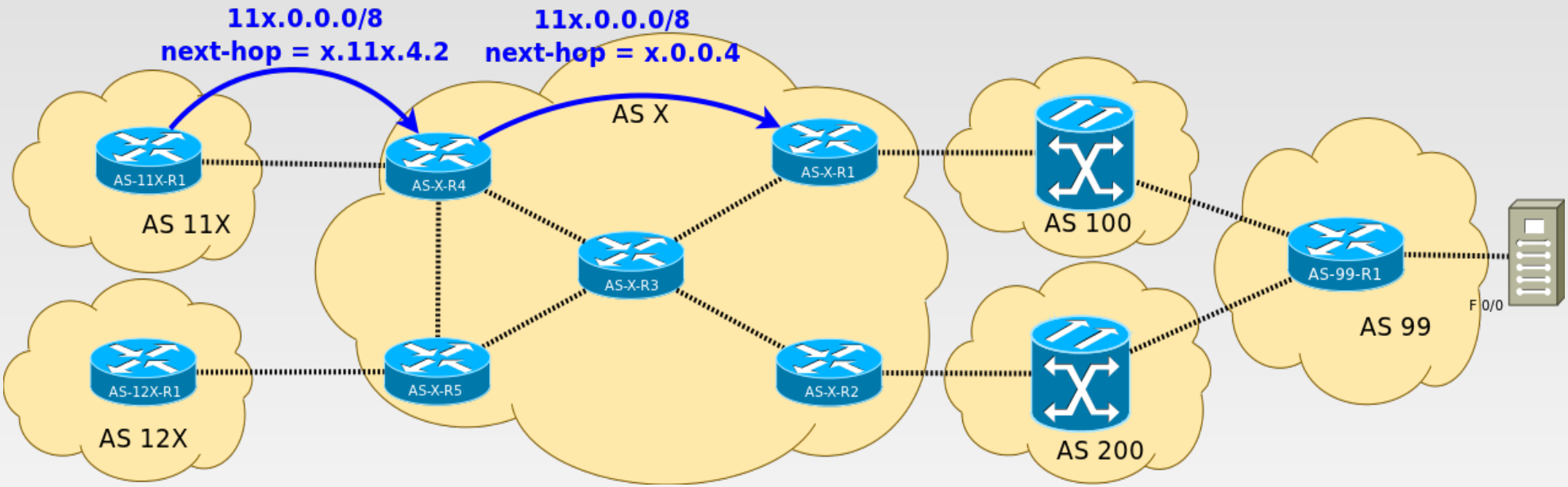
5.1 - Implementando múltiplas saídas para clientes de trânsito

- AS 14**X** anuncia seus prefixos para R4...



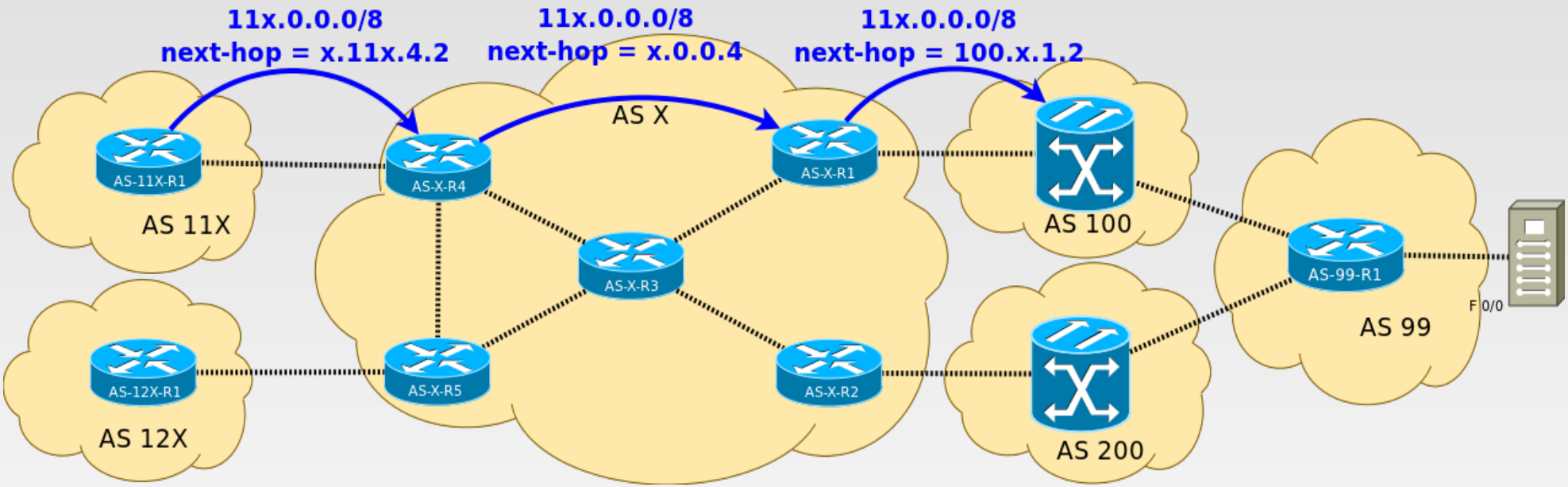
5.1 - Implementando múltiplas saídas para clientes de trânsito

- AS R4 anuncia para R1...



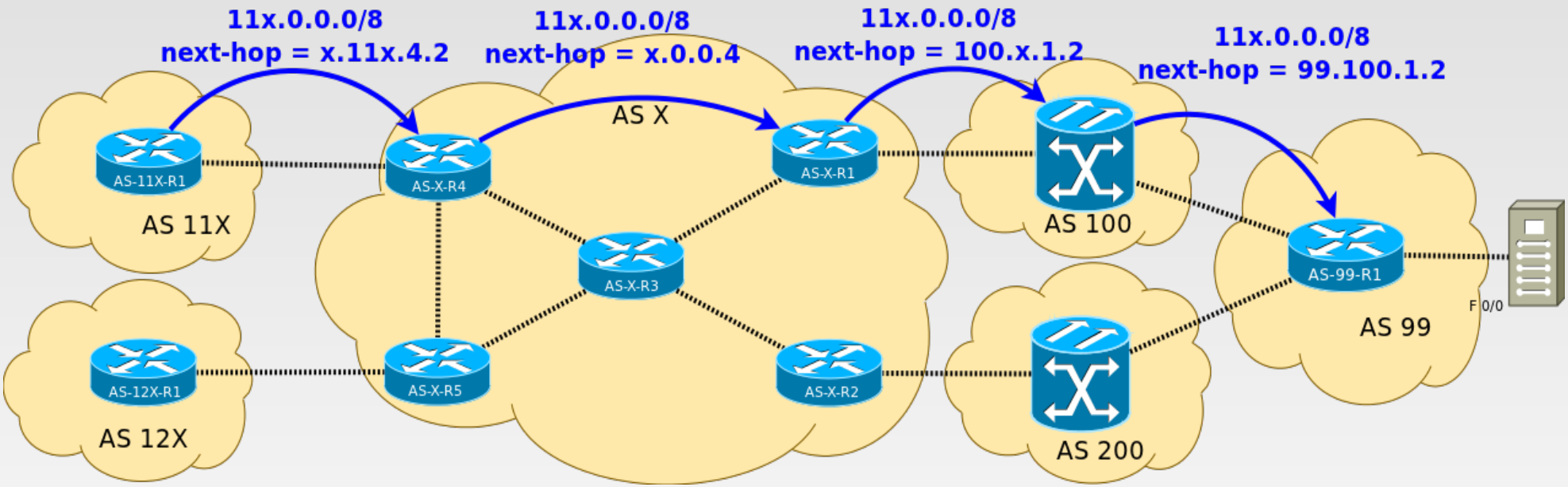
5.1 - Implementando múltiplas saídas para clientes de trânsito

- AS R1 anuncia para o UPSTREAM AS 100...



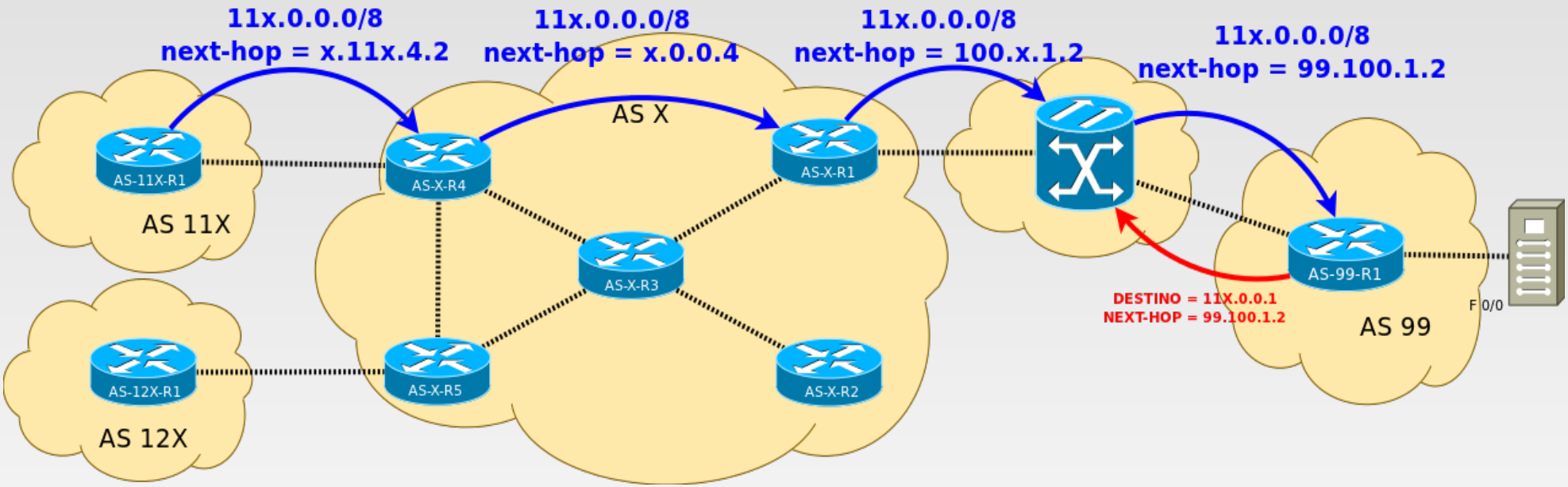
5.1 - Implementando múltiplas saídas para clientes de trânsito

- AS 100 anuncia para o AS 99...



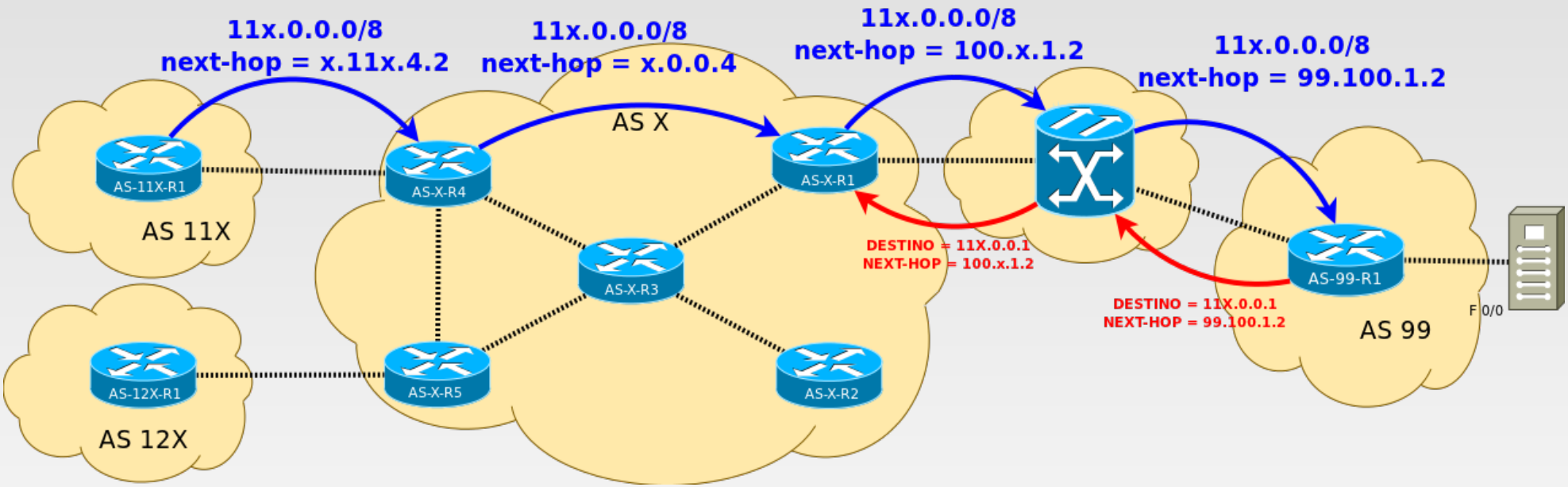
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Caminho dos pacotes: Entram no AS 100...



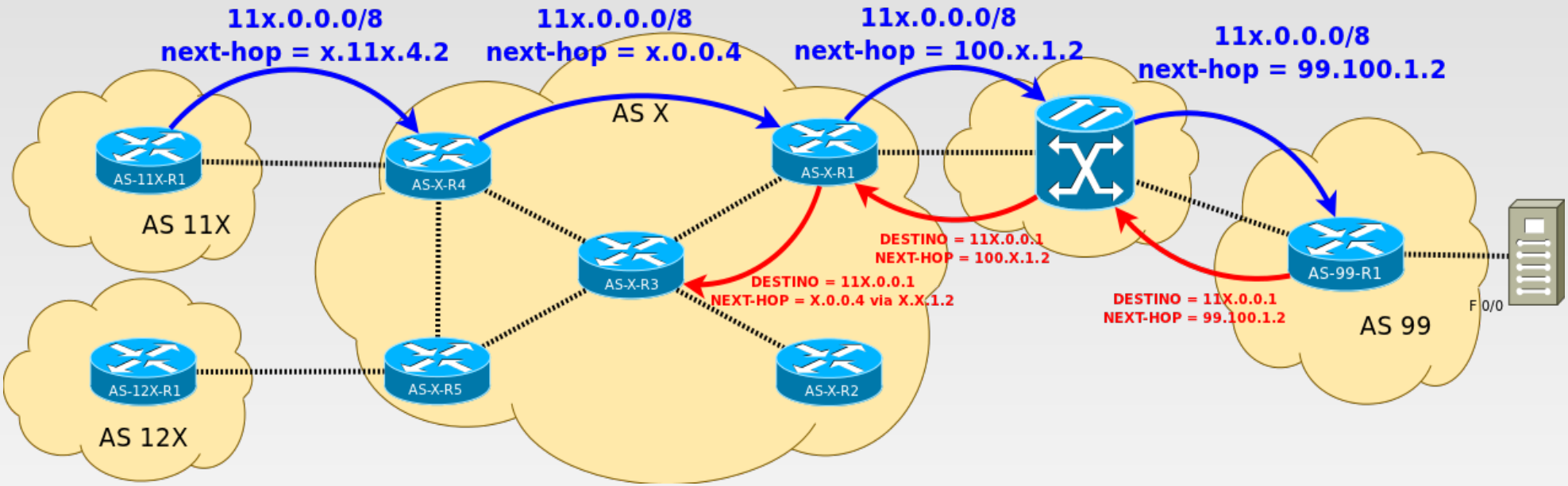
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Caminho dos pacotes: em seguida no AS por R1...



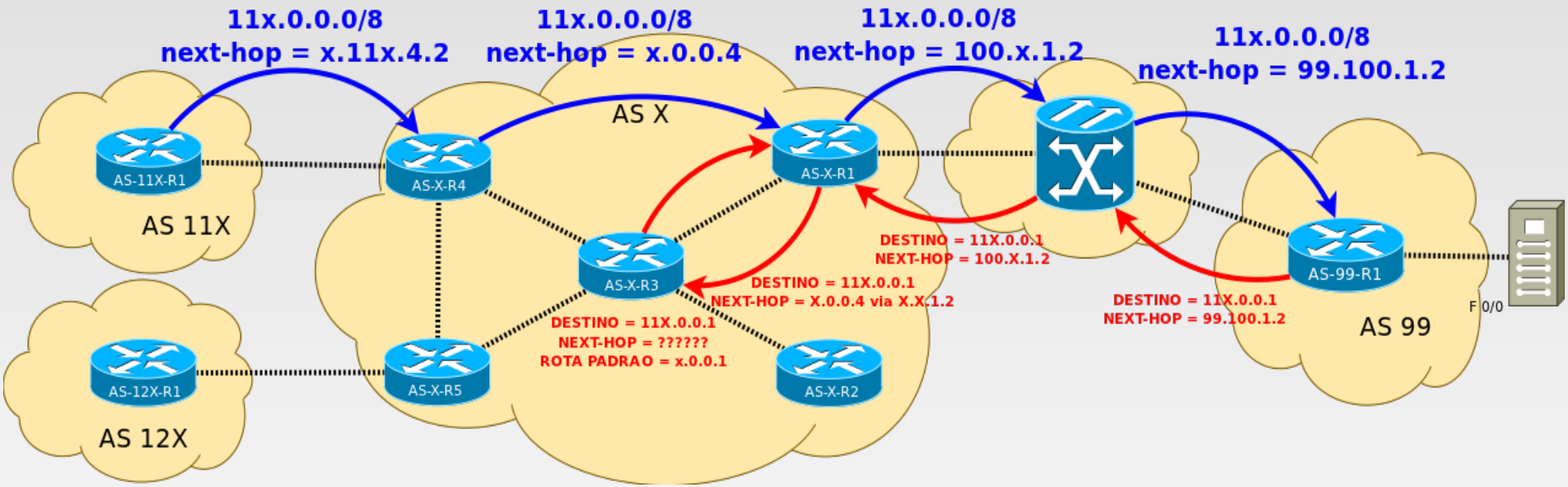
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Caminho dos pacotes: agora complica um pouco, R1 conhece rota para 11**X**.0.0.0/8 através do next-hop **X**.0.0.4 (R4), e esse next-hop é alcançável via R3 (**X**.**X**.1.2)



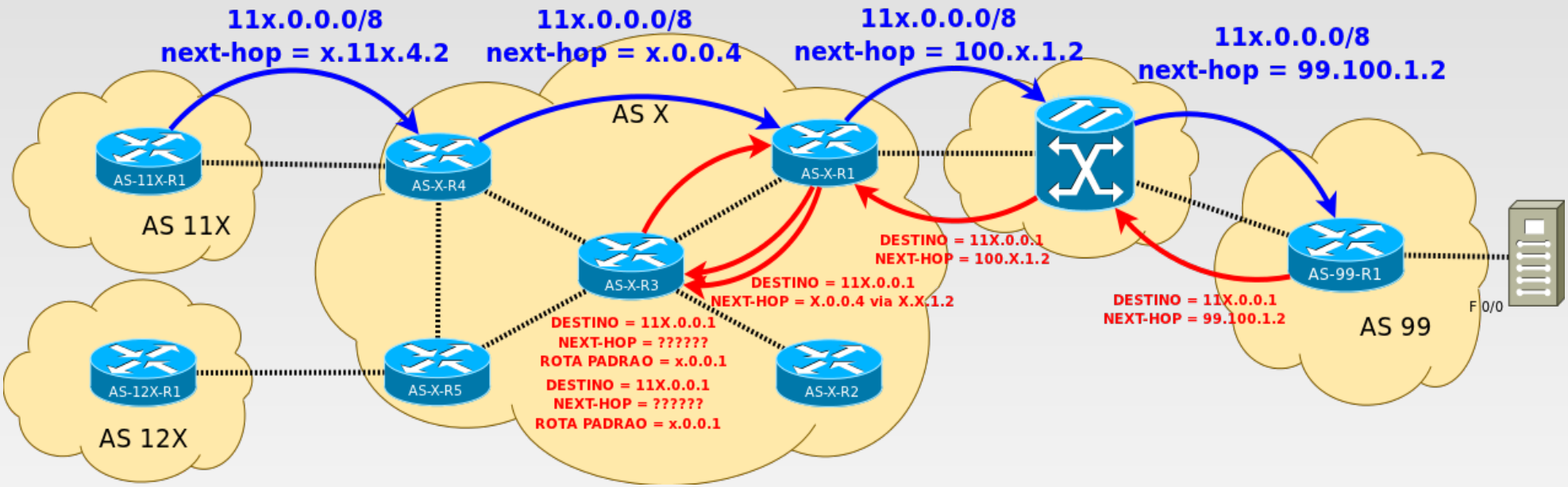
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Caminho dos pacotes: ”Ninguém avisou para R3” qualquer coisa sobre um tal AS 11**X**, o que ele faz com um pacote cujo destino não conhece? (rota padrão) O pacote é devolvido para R1



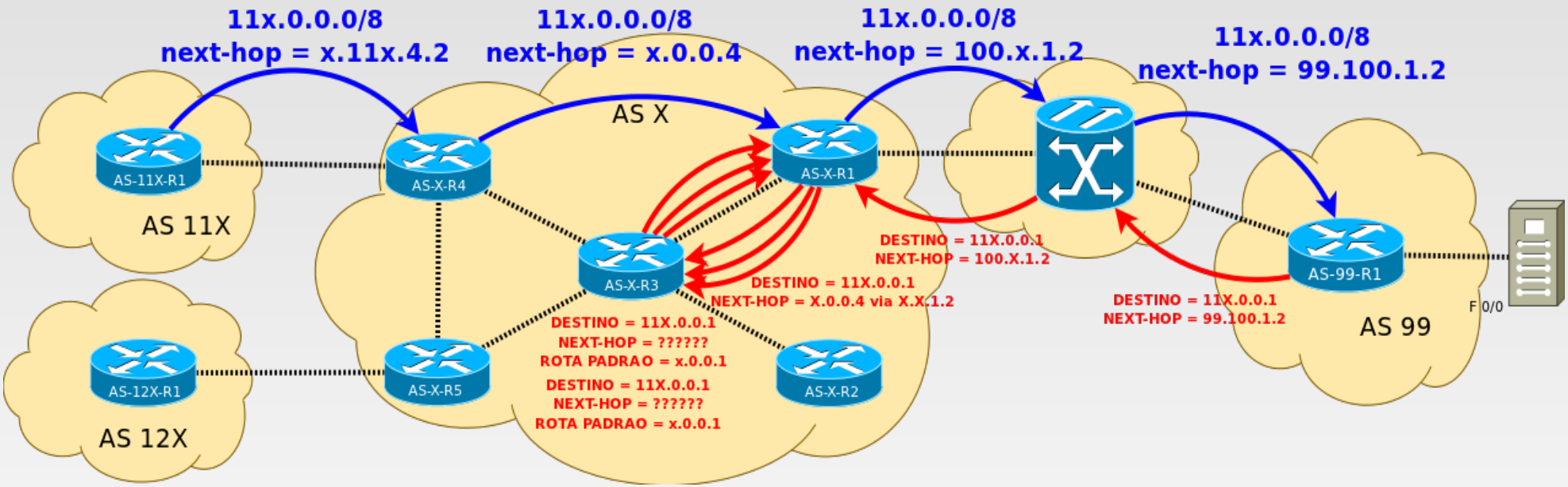
5.1 - Implementando múltiplas saídas para clientes de trânsito

- Caminho dos pacotes: Mas R1 consulta novamente sua FIB, e ”tem ceteza” que o pacote deve mesmo ser entregue para R3...



5.1 - Implementando múltiplas saídas para clientes de trânsito

- Caminho dos pacotes: E o loop se repete até o fim do TTL do pacote...



5.1 - Implementando múltiplas saídas para clientes de trânsito

- Por isso precisamos ter certeza antes de "desabilitar" o syncronization
- Veremos duas maneiras de resolver esse problema, uma delas com o syncronization HABILITADO;
- E a outra com o syncronization DESABILITADO
- Vamos à primeira!

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Distribuindo as rotas BGP através do meu IGP:

!--- primeiro vamos habilitar novamente o syncronization
AS-**X**-R1#conf t
AS-**X**-R1(config)#router bgp **X**
AS-**X**-R1(config-router)#synchronization
AS-**X**-R1(config-router)#exit
!--- agora entrar na configuração do OSPF
AS-**X**-R1(config)#router ospf 1
AS-**X**-R1(config-router)#redistribute bgp **X**
!--- habilita a distribuição por OSPF de rotas aprendias por BGP
% Only classful networks will be redistributed
AS-**X**-R1(config-router)#end
AS-**X**-R1#clear ip ospf process
!--- reinicia o processo OSPF
Reset ALL OSPF processes? [no]: yes
AS-**X**-R1#
!--- vamos repetir o mesmo processo para R2, R4 e R5

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Distribuindo as rotas BGP através do meu IGP:


```
!--- primeiro vamos habilitar novamente o syncronization
AS-X-R2#conf t
AS-X-R2(config)#router bgp X
AS-X-R2(config-router)#synchronization
AS-X-R2(config-router)#exit
!--- agora entrar na configuração do OSPF
AS-X-R2(config)#router ospf 1
AS-X-R2(config-router)#redistribute bgp X
!--- habilita a distribuição por OSPF de rotas aprendias por BGP
% Only classful networks will be redistributed
AS-X-R2(config-router)#end
AS-X-R2#clear ip ospf process
!--- reinicia o processo OSPF
Reset ALL OSPF processes? [no]: yes
AS-X-R2#
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Distribuindo as rotas BGP através do meu IGP:


```
!--- primeiro vamos habilitar novamente o syncronization
AS-X-R4#conf t
AS-X-R4(config)#router bgp X
AS-X-R4(config-router)#synchronization
AS-X-R4(config-router)#exit
!--- agora entrar na configuração do OSPF
AS-X-R4(config)#router ospf 1
AS-X-R4(config-router)#redistribute bgp X
!--- habilita a distribuição por OSPF de rotas aprendias por BGP
% Only classful networks will be redistributed
AS-X-R4(config-router)#end
AS-X-R4#clear ip ospf process
!--- reinicia o processo OSPF
Reset ALL OSPF processes? [no]: yes
AS-X-R4#
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Distribuindo as rotas BGP através do meu IGP:


```
!--- primeiro vamos habilitar novamente o syncronization
AS-X-R5#conf t
AS-X-R5(config)#router bgp X
AS-X-R5(config-router)#synchronization
AS-X-R5(config-router)#exit
!--- agora entrar na configuração do OSPF
AS-X-R5(config)#router ospf 1
AS-X-R5(config-router)#redistribute bgp X
!--- habilita a distribuição por OSPF de rotas aprendias por BGP
% Only classful networks will be redistributed
AS-X-R5(config-router)#end
AS-X-R5#clear ip ospf process
!--- reinicia o processo OSPF
Reset ALL OSPF processes? [no]: yes
AS-X-R5#
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Agora vamos verificar em R1 as rotas recebidas do AS-14**X** via R4

AS-X-R1#sh ip bgp

Network	Next Hop	Metric	LocPrf	Weight	Path
...					
*> 99.0.0.0	100.X.1.1			0	100 99 i
*> 100.0.0.0	100.X.1.1	0		0	100 i
*>i14X.0.0.0	X.0.0.4	0	100	0	14X i
*>i15X.0.0.0	X.0.0.5	0	100	0	15X i
*> 200.0.0.0	100.X.1.1			0	100 99 200 i
...					

- Agora podemos ver pelo ">" que a rota para 14**X**.0.0.0 está na FIB
- Outro detalhe é o "i" que identifica a rota como "internal"

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Detalhando os prefixos do meu DOWNSTREAM

AS-X-R1#sh ip bgp 14X.0.0.0
BGP routing table entry for 14X.0.0.0/8, version 5
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  100.X.1.1
  11X
    X.0.0.4 (metric 129) from X.0.0.4 (X.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal, **synchronized**, **best**

- Temos agora uma rota **SINCRONIZADA**, **VÁLIDA**, e **ANUNCIADA** para 100.X.1.1 (AS 100)

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Segunda maneira é desabilitando o syncronization, porém alterando os BGP peers
- Vamos refazer as configurações BGP estabelecendo R4<->R3 e R3<->R1
- Dessa forma R3 vai conhecer a rota para os DOWSTREAMS e UPSTREAMS
- Vamos desfazer todas as configuraçõs BGP...

!--- R1
AS-**X**-R1(config)#no router bgp **X**

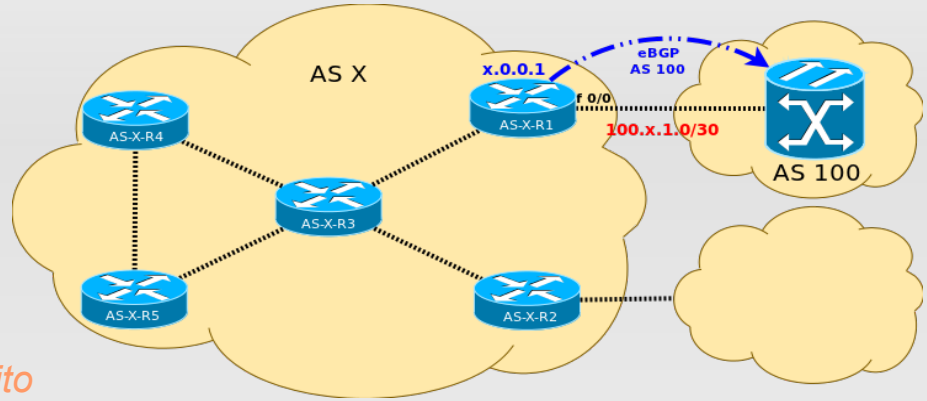
!--- R2
AS-**X**-R2(config)#no router bgp **X**

!--- R4
AS-**X**-R4(config)#no router bgp **X**

!--- R5
AS-**X**-R5(config)#no router bgp **X**

5.1 - Implementando múltiplas saídas para clientes de trânsito

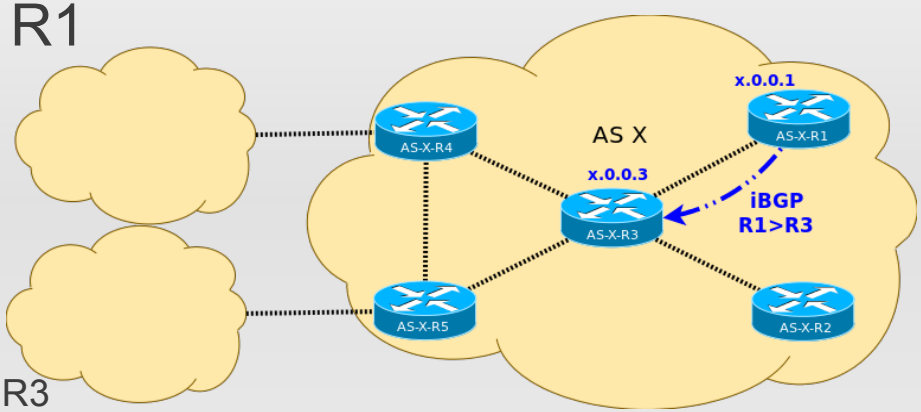
- Reestabelecendo as sessões BGP em R1
  - Primeiro eBGP entre R1 e AS 100




```
AS-X-R1#conf t
AS-X-R1(config)#router bgp X
AS-X-R1(config-router)#no synchronization
AS-X-R1(config-router)#bgp router-id X.0.0.1
!--- Dessa vez desabilitando o syncronization, e de propósito
!--- não estamos anunciando a rede x.0.0.0 em network, faremos em R3
AS-X-R1(config-router)#neighbor 100.X.1.1 remote-as 100
AS-X-R1(config-router)#neighbor 100.X.1.1 soft-reconfiguration inbound
AS-X-R1(config-router)#neighbor 100.X.1.1 route-map ANUNCIAR-AS-100 out
!--- caso surja a dúvida, não uso netx-hop-self pois com eBGP isso é nativo
!--- também nao precisamos do update-source pois utilizamos as interfaces WAN para o eBGP
AS-X-R1(config-router)#exit
!--- todas as prefix-lists e route-maps não foram removidos no comando "no router bgp X"
!--- por isso, precisei apenas declarar o route-map novamente na criação do peer
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

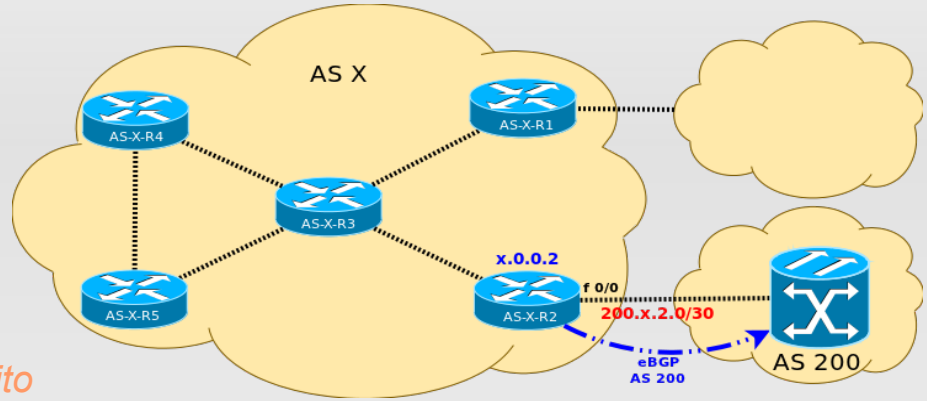
- Reestabelecendo as sessões BGP em R1
  - Em seguida iBGP entre R1 e R3




```
AS-X-R1#conf t
AS-X-R1(config)#router bgp X
AS-X-R1(config-router)#neighbor X.0.0.3 remote-as X
AS-X-R1(config-router)#neighbor X.0.0.3 description iBGP R3
AS-X-R1(config-router)#neighbor X.0.0.3 next-hop-self
AS-X-R1(config-router)#neighbor X.0.0.3 update-source loopback 0
AS-X-R1(config-router)#neighbor X.0.0.3 soft-reconfiguration inbound
AS-X-R1(config-router)#exit
!--- não vamos utilizar nenhum filtro de in-out para a sessão com R3
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

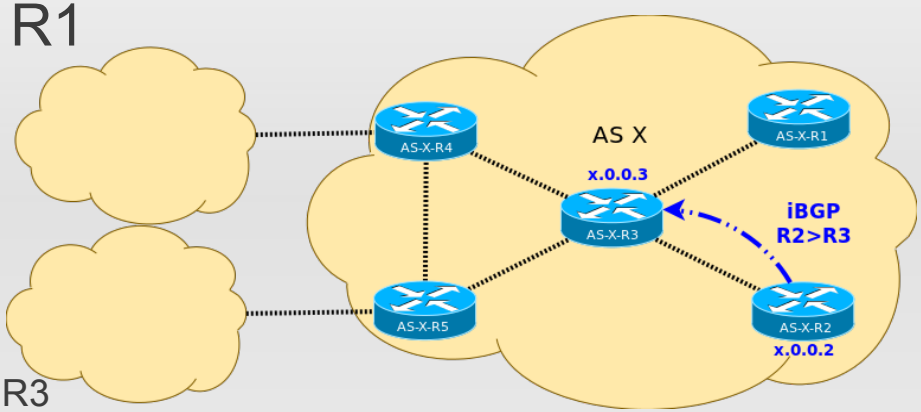
- Reestabelecendo as sessões BGP em R2
  - Primeiro eBGP entre R2 e AS 200




```
AS-X-R2#conf t
AS-X-R2(config)#router bgp X
AS-X-R2(config-router)#no synchronization
AS-X-R2(config-router)#bgp router-id X.0.0.1
!--- Dessa vez desabilitando o syncronization, e de propósito
!--- não estamos anunciando a rede x.0.0.0 em network, faremos em R3
AS-X-R2(config-router)#neighbor 200.X.2.1 remote-as 200
AS-X-R2(config-router)#neighbor 200.X.2.1 soft-reconfiguration inbound
AS-X-R2(config-router)#neighbor 200.X.2.1 route-map ANUNCIAR-AS-200 out
!--- caso surja a dúvida, não uso netx-hop-self pois com eBGP isso é nativo
!--- também nao precisamos do update-source pois utilizamos as interfaces WAN para o eBGP
AS-X-R2(config-router)#exit
!--- todas as prefix-lists e route-maps não foram removidos no comando "no router bgp X"
!--- por isso, precisei apenas declarar o route-map novamente na criação do peer
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

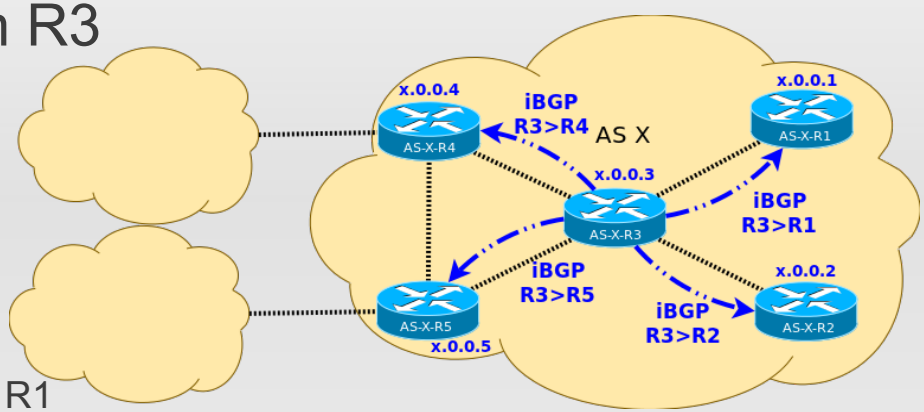
- Reestabelecendo as sessões BGP em R1
  - Em seguida iBGP entre R1 e R3




```
AS-X-R2#conf t
AS-X-R2(config)#router bgp X
AS-X-R2(config-router)#neighbor X.0.0.3 remote-as X
AS-X-R2(config-router)#neighbor X.0.0.3 description iBGP R3
AS-X-R2(config-router)#neighbor X.0.0.3 next-hop-self
AS-X-R2(config-router)#neighbor X.0.0.3 update-source loopback 0
AS-X-R2(config-router)#neighbor X.0.0.3 soft-reconfiguration inbound
AS-X-R2(config-router)#exit
!--- não vamos utilizar nenhum filtro de in-out
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

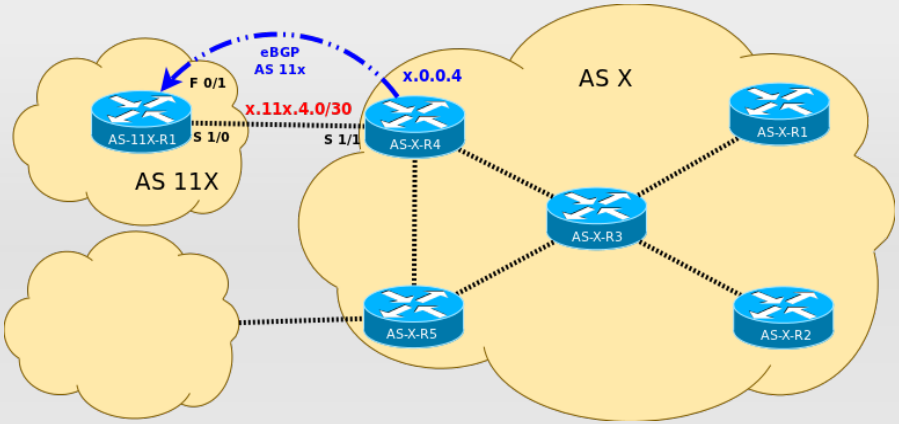
- Reestabelecendo as sessões iBGP em R3




```
AS-X-R3#conf t
AS-X-R3(config)#router bgp X
AS-X-R3(config-router)#no synchronization
AS-X-R3(config-router)#network X.0.0.0
AS-X-R3(config-router)#bgp router-id X.0.0.3
!--- iBGP com R1
AS-X-R3(config-router)#neighbor X.0.0.1 remote-as X
AS-X-R3(config-router)#neighbor X.0.0.1 description iBGP R1
AS-X-R3(config-router)#neighbor X.0.0.1 update-source loopback 0
AS-X-R3(config-router)#neighbor X.0.0.1 soft-reconfiguration inbound
!--- iBGP com R2
AS-X-R3(config-router)#neighbor X.0.0.2 remote-as X
AS-X-R3(config-router)#neighbor X.0.0.2 description iBGP R2
AS-X-R3(config-router)#neighbor X.0.0.2 update-source loopback 0
AS-X-R3(config-router)#neighbor X.0.0.2 soft-reconfiguration inbound
!--- iBGP com R4
AS-X-R3(config-router)#neighbor X.0.0.4 remote-as X
AS-X-R3(config-router)#neighbor X.0.0.4 description iBGP R4
AS-X-R3(config-router)#neighbor X.0.0.4 update-source loopback 0
AS-X-R3(config-router)#neighbor X.0.0.4 soft-reconfiguration inbound
!--- iBGP com R5
AS-X-R3(config-router)#neighbor X.0.0.5 remote-as X
AS-X-R3(config-router)#neighbor X.0.0.5 description iBGP R5
AS-X-R3(config-router)#neighbor X.0.0.5 update-source loopback 0
AS-X-R3(config-router)#neighbor X.0.0.5 soft-reconfiguration inbound
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

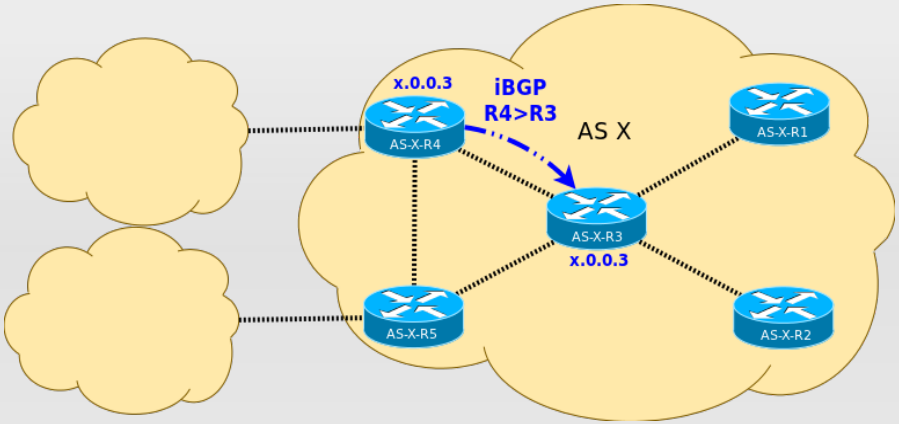
- Configurando eBGP com o downstream:
  AS 14**X** em R4:




```
AS-X-R4#conf t
AS-X-R4(config)#router bgp X
AS-X-R4(config-router)#bgp router-id X.0.0.4
AS-X-R4(config-router)#no synchronization
!--- Dessa vez desabilitando o syncronization, e de propósito
!--- não estamos anunciando a rede x.0.0.0 em network, faremos em R3
AS-X-R4(config-router)#neighbor X.14X.4.2 remote-as 14X
AS-X-R4(config-router)#neighbor X.14X.4.2 description eBGP-downstream-AS-14X
AS-X-R4(config-router)#neighbor X.14X.4.2 soft-reconfiguration inbound
AS-X-R4(config-router)#neighbor X.14X.4.2 prefix-list BLOCOS-AS-14X in
!--- caso surja a dúvida, não uso netx-hop-self pois com eBGP isso é nativo
!--- também nao precisamos do update-source pois utilizamos as interfaces WAN para o eBGP
AS-X-R4(config-router)#exit
!--- todas as prefix-lists e route-maps não foram removidos no comando ”no router bgp X”
!--- por isso, precisamos apenas declarar o route-map novamente na criação do peer
```

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Configurando iBGP com R3

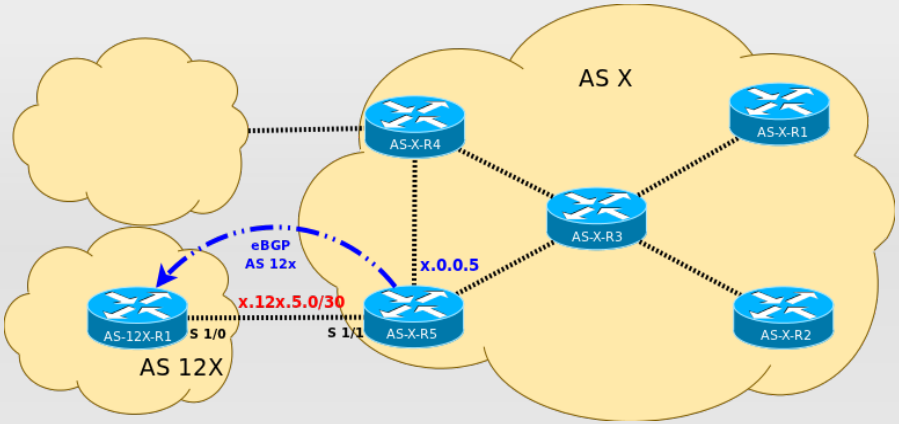


AS-**X**-R4#conf t
AS-**X**-R4(config)#router bgp **X**

AS-**X**-R4(config-router)#neighbor **X**.0.0.3 remote-as **X**
AS-**X**-R4(config-router)#neighbor **X**.0.0.3 description iBGP R3
AS-**X**-R4(config-router)#neighbor **X**.0.0.3 next-hop-self
AS-**X**-R4(config-router)#neighbor **X**.0.0.3 update-source loopback 0
AS-**X**-R4(config-router)#neighbor **X**.0.0.3 soft-reconfiguration inbound
AS-**X**-R4(config-router)#exit

5.1 - Implementando múltiplas saídas para clientes de trânsito

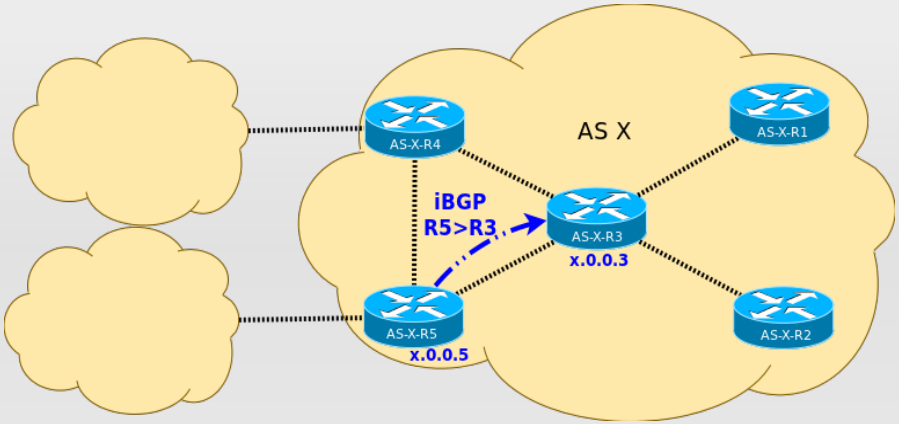
- Configurando eBGP com o downstream:
  AS 15**X** em R5:



AS-**X**-R5#conf t
AS-**X**-R5(config)#router bgp **X**
AS-**X**-R5(config-router)#bgp router-id **X**.0.0.5
AS-**X**-R5(config-router)#no synchronization
!--- dessa vez desabilitando o syncronization, e de propósito
!--- não estamos anunciando a rede x.0.0.0 em network, faremos em R3
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 remote-as 12**X**
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 description eBGP-downstream-AS-15**X**
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 soft-reconfiguration inbound
AS-**X**-R5(config-router)#neighbor **X**.15**X**.5.2 prefix-list BLOCOS-AS-15**X** in
!--- caso surja a dúvida, não uso netx-hop-self pois com eBGP isso é nativo
!--- também nao precisamos do update-source pois utilizamos as interfaces WAN para o eBGP
AS-**X**-R5(config-router)#exit
!--- todas as prefix-lists e route-maps não foram removidos no comando "no router bgp X"
!--- por isso, precisamos apenas declarar o route-map novamente na criação do peer

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Configurando iBGP com R3



AS-**X**-R5#conf t
AS-**X**-R5(config)#router bgp **X**

AS-**X**-R5(config-router)#neighbor **X**.0.0.3 remote-as **X**
AS-**X**-R5(config-router)#neighbor **X**.0.0.3 description iBGP R3
AS-**X**-R5(config-router)#neighbor **X**.0.0.3 next-hop-self
AS-**X**-R5(config-router)#neighbor **X**.0.0.3 update-source loopback 0
AS-**X**-R1(config-router)#neighbor **X**.0.0.3 soft-reconfiguration inbound
AS-**X**-R5(config-router)#exit

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Verificando as mudanças começando por R1

AS-X-R1#sh ip bgp
BGP table version is 2, local router ID is X.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*>iX.0.0.0	X.0.0.3	0	100	0	i
*> 100.0.0.0	100.X.1.1	0		0	100 i
*> 99.0.0.0	100.X.1.1	0		0	100 99 i

- Ainda não temos a rota anunciada pelo DOWNSTRASM AS 14X na FIB de R1, isso porque R3 não está anunciando!
- Por que isso acontece?

5.1 - Implementando múltiplas saídas para clientes de trânsito

- BGP e o SPLIT HORIZON

- Desde o tempo do RIP, os protocolos de vetor distância sabem que não há sentido em propagar uma rota para a mesma interface de onde aprendeu, e com o BGP é parecido

- No iBGP é indispensável* que haja fullmesh de sessões BGP (todos estabelecendo com todos) e sendo assim, um router não anuncia por iBGP uma rota que foi aprendida também por iBGP, pois entende que aquele router que me enviou a rota já está enviando diretamente para todos os routers do iBGP

*há um mecanismo que dispensa essa necessidade

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Em resumo, um router não anuncia para outro router do mesmo AS qualquer coisa que tenha aprendido também via iBGP

- Ou seja, R3 não vai anunciar para R1 as rotas aprendidas de R4

5.2
UTILIZANDO ROUTE-REFLECTOR

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

5.2 - Utilizando Route-reflector

- Em tese, de acordo com as regras do split horizon deveríamos estabelecer as seguintes sessões BGP (full mesh):

  R1<->R2, R1<->R3, R1<->R4, R1<->R5

  R2<->R3, R2<->R4, R1<->R5

  R3<->R4, R3<->R5

  R4<->R5

  Total=10

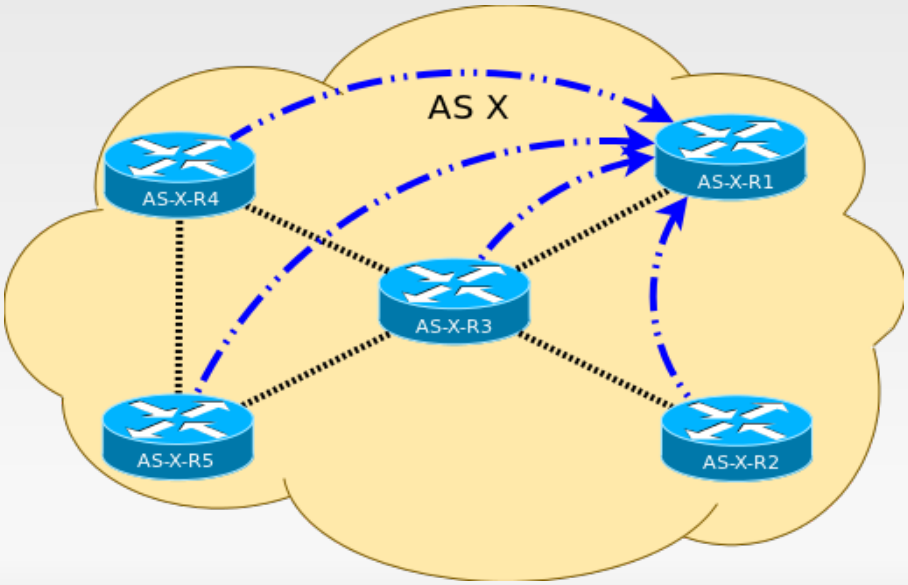
- O route-reflector basicamente dispensa a necessidade de iBGP full-mesh, pois remove a restrição no repasse de rotas iBGP para outro neighbor iBGP
- É altamente recomendado quando o número de routers é maior que 5
  *evita um número muito grande de sessões iBGP

5.2 - Utilizando Route-reflector

- Nessa topologia de rede em particular, podemos optar por 3 modelos:

  1- R1 como Route-reflector estabelecendo com R2, R3, R4 e R5 (4 sessões)

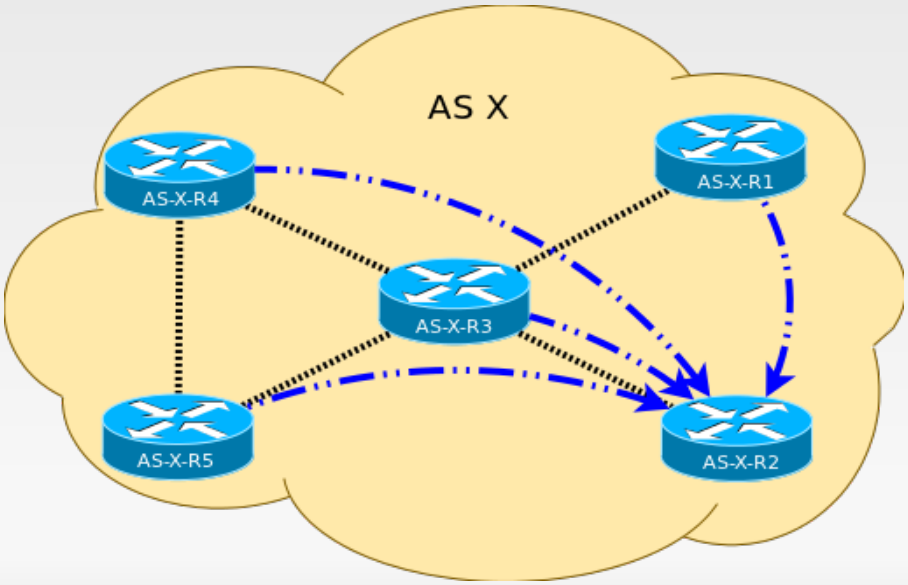
- Pontos críticos de falha: falha em R3 e/ou R1 e/ou link R3<->R1 = 3



5.2 - Utilizando Route-reflector

2- R2 como Route-reflector estabelecendo com R1, R3, R4 e R5 (4 sessões)

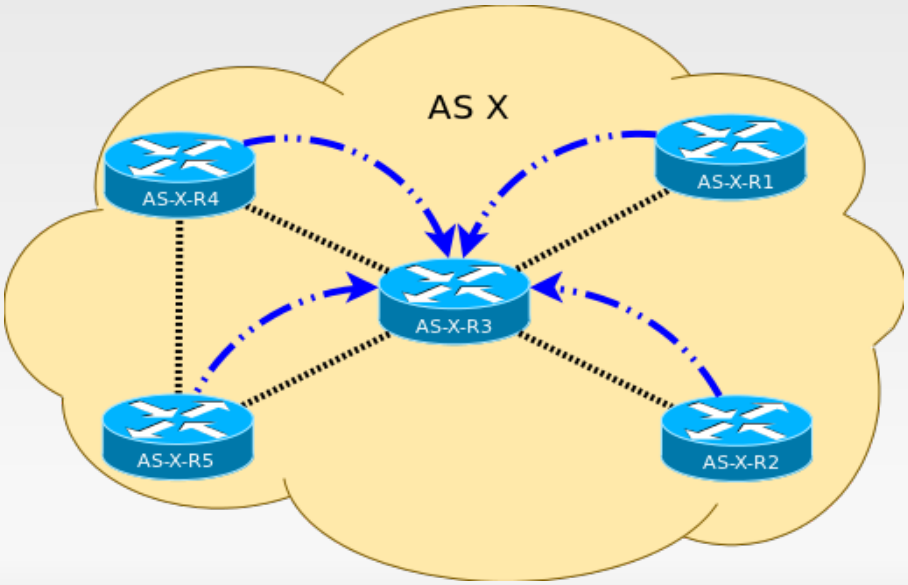
- Pontos críticos de falha: falha em R3 e/ou R2 e/ou link R3<->R2 = 3



5.2 - Utilizando Route-reflector

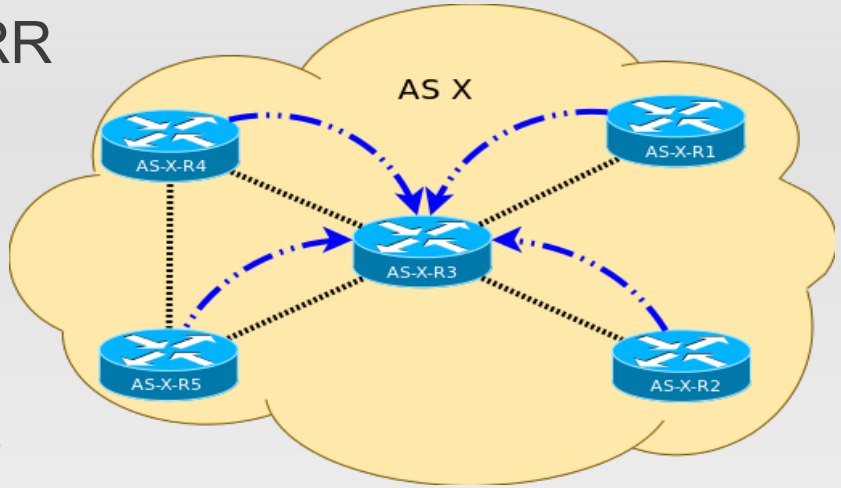
3- R3 como Route-reflector estabelecendo com R1, R2, R4 e R5 (4 sessões)

- Pontos críticos de falha: falha em R3= 1



5.1 - Implementando múltiplas saídas para clientes de trânsito

- Obviamente já optamos pela opção número 3
  - Vamos acrescentar apenas a opção RR




```
AS-X-R3#conf t
AS-X-R3(config)#router bgp X
!--- iBGP com R1
AS-X-R3(config-router)#neighbor X.0.0.1 route-reflector-client
!--- iBGP com R2
AS-X-R3(config-router)#neighbor X.0.0.2 route-reflector-client
!--- iBGP com R4
AS-X-R3(config-router)#neighbor X.0.0.4 route-reflector-client
!--- iBGP com R5
AS-X-R3(config-router)#neighbor X.0.0.5 route-reflector-client
```

5.2 - Utilizando Route-reflector

- É importante deixar claro que diferentemente do que o nome possa sugerir, "Route-reflector" não significa "refletir a rota" e preservar o next-hop original, isso é uma característica do próprio iBGP. Route-Reflector significa anunciar via iBGP rotas aprendidas também por iBGP

5.3
UTILIZANDO PEER-GROUPS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

5.3 - Utilizando Peer groups

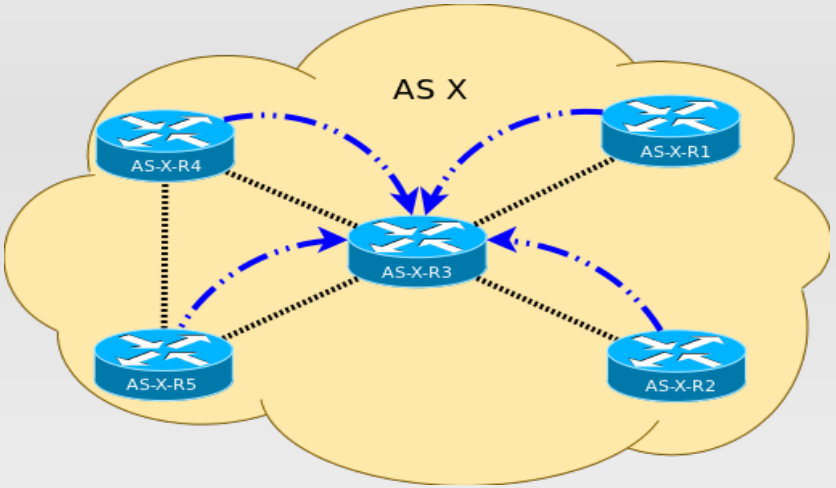
- Em nossa topologia, ficamos diante de uma situação:

- R3 possui vários neighbors com configurações semelhantes, exigindo configuraçõs repetitivas e sujeitas à erros

- O recurso "peer-group" me permite realizar um padrão para os neighbors semelhantes diminuindo drasticamente a quantidade de linhas de configuração e "show run"

5.1 - Implementando múltiplas saídas para clientes de trânsito

- Vamos praticar reconfigurando R3 com peer-groups:



!--- Primeiro vamos resetar as config de BGP
AS-**X**-R3#conf t
AS-**X**-R3(config)#no router bgp **X**

!--- depois as configurações básicas
AS-**X**-R3(config)#router bgp **X**
AS-**X**-R3(config-router)#no synchronization
AS-**X**-R3(config-router)#bgp router-id **X**.0.0.3
AS-**X**-R3(config-router)#network **X**.0.0.0
!--- Vamos declarar um peer-group chamado IBGP
AS-**X**-R3(config-router)#neighbor **IBGP** peer-group
AS-**X**-R3(config-router)#neighbor **IBGP** remote-as **X**
AS-**X**-R3(config-router)#neighbor **IBGP** update-source loopback 0
AS-**X**-R3(config-router)#neighbor **IBGP** soft-reconfiguration inbound
AS-**X**-R3(config-router)#neighbor **IBGP** route-reflector-client
!--- Vamos adicionar os neighbors que fazem parte desse grupo
AS-**X**-R3(config-router)#neighbor **X**.0.0.1 peer-group **IBGP**
AS-**X**-R3(config-router)#neighbor **X**.0.0.2 peer-group **IBGP**
AS-**X**-R3(config-router)#neighbor **X**.0.0.4 peer-group **IBGP**
AS-**X**-R3(config-router)#neighbor **X**.0.0.5 peer-group **IBGP**

5.3 - Utilizando Peer groups

- Com peer-group fica muito mais simples adicionar remover configurações de neighbors para todo o grupo em apenas um comando, por exemplo:

```
!--- apenas exemplo, não efetue essa configuração
router(config-router)#neighbor IBGP prefix-list FILTRO-GERAL-DE-ENTRADA in
router(config-router)#neighbor IBGP prefix-list FILTRO-GERAL-DE-SAIDA out
!--- apenas exemplo, não efetue essa configuração
```

- Todas as sessões BGP dentro do peer-group passariam pelos mesmos filtros de entrada e de saída

```
!--- A mesma simplicidade acontece na hora de um "clear" (route refresh)
AS-X-R3#clear ip bgp peer-group IBGP soft
!--- mas vale lembrar que nada impede que seja efetuado um clear individual
AS-X-R3#clear ip bgp X.0.0.1 out
```

5.3 - Utilizando Peer groups

- Finalizando o LAB:
- Acessar o Looking Glass e verificar se esse alcança o AS do grupo e todos os seus respectivos DOWNSTREAMS

telnet 99.99.99.1 ...

5.4
LOCAL PREFERENCE x WEIGHT NO iBGP

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

5.4 - Local Preference x Weight no iBGP

- Características:

Weight:

- Proprietário Cisco (embora implementado por Mikrotik e Quagga)
- Primeiro critério na escolha de rotas (maior weight)
- Não é considerado um atributo e seu valor pode ser de 0 a 65535
- Os valores de weight não são transitivos (mesmo dentro do AS)

Local Preference:

- Conhecido em todas as implementações de BGP
- Segundo critério depois do weight (maior local pref)
- Seu valor padrão é 100 e pode variar de 0 a 4294967295
- Os valores alterados são propagados por todo o iBGP

5.4 - Local Preference x Weight no iBGP

- Como vimos nos labs anteriores, posso atribuir um valor de weight 10 no comando "neighbor x.x.x.x weight 10" para todas as rotas recebidas, ou manualmente apenas para determinados prefixos utilizando o route-map. Vamos alterar o weight apenas de um determinado prefixo (99.0.0.0/8)

AS-**X**-R1#conf t
AS-**X**-R1(config)#ip prefix-list BLOCOS-AS-99 permit 99.0.0.0/8
!--- adiciona o classe A e sub-redes 90.0.0.0 na prefix-list BLOCOS-AS-99
AS-**X**-R1(config)#router bgp **X**
AS-**X**-R1(config-router)#neighbor 100.**X**.1.1 route-map AS-100-IN in
!--- declara o route-map AS-100-IN para filtrar anuncios de input do AS 100
AS-**X**-R1(config-router)#exit
AS-**X**-R1(config)#route-map AS-100-IN permit 10
AS-**X**-R1(config-route-map)#match ip address prefix-list BLOCOS-AS-99
AS-**X**-R1(config-route-map)#set weight 10
!--- estabelece os prefixos da lista BLOCOS-AS-99 como condição para receber a ação permir
!--- e a modificação opcional que atribui um valor 10 de weight
*AS-**X**-R1(config-route-map)#exit*
*AS-**X**-R1(config)#route-map AS-100-IN permit 999*
!--- não podemos esquecer de adicionar um route-map ”vazio” no final (numero alto), caso contrário
!--- os prefixos do AS 99 reeceberiam a ação desejada, porém nenhum outro prefixo seria aceito
*AS-**X**-R1(config-route-map)#end*
*AS-**X**-R1#clear ip bgp 100.**X**.1.1 soft*

5.4 - Local Preference x Weight no iBGP

- Conferindo a modificação:

AS-X-R1#sh ip bgp

```
...
   Network          Next Hop            Metric LocPrf Weight Path

*> 99.0.0.0         100.X.1.1                0             10 100 99 i
*> 100.0.0.0       100.X.1.1                0              0 100 i
*> 200.0.0.0       100.X.1.1                0              0 100 99 200 i
...
```

- Apenas o prefixo 99.0.0.0 recebeu weight 10
- Verifiquem a mesma rota em R2, R3,R4 ou R5 o valor de weight será 0, já que esse não é propagado através do iBGP

5.4 - Local Preference x Weight no iBGP

- Já com o local preference é diferente, o valor que configurarmos em R1 permanecerá o mesmo em todos os routers do AS que receberem esse prefixo através de iBGP
- **PS:** Caso algum router aplique routemap que altere o local preference do prefixo 99.0.0.0 ou de todos os prefixos recebidos, esse valor é substituído pelo atual
- A maneira mais indicada de priorizar rotas dentro do iBGP é utilizando o localpreference
- o weight deve ser usado apenas em casos muito específicos onde não haja a intenção de que os outros routers do AS escolham essa mesma rota

6.3 - BGP Communities

- Solução:

AS-**X**-R1#conf t
AS-**X**-R1(config)#ip as-path access-list 44 permit _1[4|5]**X**$
*!--- lembrem que **X** é o AS do grupo*

AS-**X**-R1(config)#route-map ANUNCIAR-AS-100 permit 5
!--- vamos utilizar o permit 5 para garantir que esse filtro fique acima
AS-**X**-R1(config-route-map)#match as-path 44
AS-**X**-R1(config-route-map)#set community NO-EXPORT
AS-**X**-R1(config-route-map)#end
AS-**X**-R1#clear ip bgp * soft

- Agora verifiquem o resultado com tracerouters a partir de AS-100-R1 e AS-99-R1

5.4 - Local Preference x Weight no iBGP

- *Desafio:
- utilizando os recursos vistos no curso vamos configurar os routers do AS de maneira que atenda todos os seguintes requisitos:
  - Todo o tráfego outbound (upload) deve utilizar o link do AS 100, exceto com destino ao AS 200 e seus DOWNSTREAMS
  - Simulando que o link do AS **100** está sobrecarregado, o tráfego inbound (download) do DOWNSTREAM AS 15**X** (apenas) deve chegar pelo AS 200 quando a origem for o AS 99 (internet)
  - Apenas R2 deve alcançar o AS 99 através do link com o AS 200

*LAB opcional

6.1
BGP COMMUNITIES

6.1 - BGP Communities

- Communities são atributos que são preservados em anúncios entre AS's externos e internos
- São um tipo de carimbo inserido nos updates em forma de string
- Uma community é um atributo opcional, e sendo assim nada obriga que os fabricantes implementem no BGP de suas caixas, ou mesmo que um AS replique para seus peers o mesmo valor de community recebido por um de seus clientes

6.1 - BGP Communities

- Os modelos mais novos suportam o "novo formato" usualmente utilizado como ASN:comunity, por exemplo: 28135:110

- Porém há algumas communities padronizadas pela RFC 1997 e são padrão nos routers Cisco:
  - INTERNET
  - NO-EXPORT
  - NO-ADVERTISE
  - LOCAL-AS
  - NO-ADVERTISE-SUBCONFED

6.1 - BGP Communities

- **NO-EXPORT** por padrão faz o router receptor exportar o prefixo para neighbors iBGP, mas não para eBGP. É útil quando faço anúncios mais especícos para a operadora A e menos específicos para operadora B. Normalmente, a operadora B precisaria passar pela operadora A para chegar ao prefixo "específico" mesmo estabelecendo BGP diretamente com meu AS.

  Enviando o mesmo bloco específico, com a community NO-EXPORT à operadora A, essa passaria a me alcançar diretamente, ao mesmo tempo que não exportaria esses prefixos.

6.3 - BGP Communities

- **NO-ADVERTISE** por padrão faz o router receptor não exportar esse prefixo para nenhum neighbor, seja ele eBGP ou iBGP

- **NO-ADVERTISE-SUBCONFED** usada para evitar que o prefixo seja exportado para AS's confederados

- **LOCAL-AS** Usada para evitar que os prefixos marcados sejam exportados para fora do AS ou confederação, geralmente usado em filtros de entrada

- **INTERNET** Na prática é a community padrão para anuncios externos, é o mesmo que não haver nenhuma

6.3 - BGP Communities

- LAB 1: Anunciar os prefixos de um dos AS's clientes com a community NO-EXPORT para dentro do AS do grupo:

```
AS-14X-R1#conf t
AS-14X-R1(config)#router bgp 14X
AS-14X-R1(config-router)#neighbor X.14X.4.1 send-community both

As-14X-R1(config)#ip bgp-community new-format

AS-14X-R1(config)#route-map AS-X-OUT permit 10
AS-14X-R1(config-route-map)#set community no-export
```

- Verifiquem nos routers R1 e R2 se os prefixos marcados como "NO-EXPORT" estão sendo anunciados para os AS's 100 e 200

6.3 - BGP Communities

- Antes de iniciar os LABs desse capítulo é **obrigatório habilitar suporte ao envio de commuinities** em TODOS os neighbors
- Em sequida habilitar o formato novo formato xxx:yyy para o tratamento de communities:

```
router#conf t
router(config)#router bgp X
router(config-router)#neighbor A.B.C.D send-community both
router(config-router)#exit
router(config)#ip bgp-community new-format
```

6.3 - BGP Communities

- LAB 2 opcional: verificar a diferença enviando a community NO-ADVERTISE

- LAB 3 :
  - Utilizar communities PADRÃO e fazer com que o **DOWNLOAD** dos AS's 14**X** e 15**X** cheguem pelo link do AS 200 quando a origem for o AS 99 (internet)
  - Os routers do AS 100 deverão alcançar os DOWNSTREAMS do grupo pelo link direto sem ter que ”dar a volta” pelo AS 99

6.4
MANIPULANDO TRÂNSITO (NACIONAL / INTERNACIONAL) COM COMMUNITIES

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

6.4 - Manipulando trânsito (nacional/internacional) com communities

- É muito pequena a quantidade de communities padrão, podemos fazer pouquíssimas coisas com elas
- Porém há uma grande quantidade communities numéricas que podem ser utilizadas.
- É importante saber que essas communities numéricas por padrão não influenciam em nada no tratamento das rotas
- É preciso uma configuração prévia ação para cada communitie

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Também não há nenhum padrão para essas ações (pelo menos nenhum padrão RFC)
- Entretanto muitos AS's implementaram padrões extremamente criativos e funcionais
- O conhecimento desses padrões é uma ferramenta chave para resolver rapidamente incidentes do dia a dia
- Vamos ver nos próximos slides uma série de communities numéricas adotadas por algumas operadoras

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Brasil Telecom

8167:90 Set Local Preference 90
8167:100 Set Local Preference 100
8167:110 Set Local Preference 110
* (default é Local Preference 200)

8167:3xy controle de anúncios

x: ação { 0 = não anuncia, 1,2,3 = insere x prepends)
y: peerings { 0=demais peerings, 1=internacionais, 2=Embratel, 3=Intelig, 4=Telemar, 5=Telefônica }
(exemplo: 312 = muda para 1 prepend o anúncio para a Embratel)

* (default é anunciar sem prepends para todos os peerings)

8167:555 => não exporta fora da Brasil Telecom.
8167:557 => não anuncia para sites internacionais.

8167:666 => serão injetados no BlackHole da Brasil Telecom.
8167:777 => somente anuncia para clientes da Brasil Telecom.

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Embratel

4230:120 - Local Preference - marcar a rota como localpreference 120
4230:10000 - Blackhole - bloqueia todo o tráfego para a rede/endereco
4230:10002 - Blackhole - filtra o tráfego internacional nos provedores que proveem alguma especie de blackhole.
4230:10004 - Blackhole - filtra nos roteadores da Embratel nos EUA o trafego destinado a rede/endereco anunciado

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Global Crossing

3549:100 - set local preference 100
3549:200 - set local preference 200
3549:275 - set local preference 275
3549:300 - set local preference 300
3549:350 - set local preference 350

3549:600 - Deny inter-continental export of tagged prefix [iBGP].
3549:666 - Deny inter-as export of tagged prefix (carry on AS 3549 only) [eBGP]

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Uma série de communities complexas possibilitam o tratamento do prepend para cada um dos peers da GLBX 3549:8...

ASN	Peer	No Export	Prepend+1	Prepend+2	Prepend+3
209	Qwest	8010	8011	8012	8013
701	MCI	8030	8031	8032	8033
1239	Sprint	8060	8061	8062	8063
1299	TeliaSonera	8250	8251	8252	8253
1668	AOL	8070	8071	8072	8073
2497	JPNIC	8080	8081	8082	8083
2516	KDDI	8100	8101	8102	8103
2914	NTT Verio	8120	8121	8122	8123
3257	Tiscali	8240	8241	8242	8243
3300	InfoNet Europe	8130	8131	8132	8133
3303	Swisscom	8140	8141	8142	8143
3320	T-Systems/DTAG	8150	8151	8152	8153
3356	Level 3	8160	8161	8162	8163
3561	Savvis	8170	8171	8172	8173
4134	ChinaNet	8230	8231	8232	8233
5511	OpenTransit	8190	8191	8192	8193
6461	AboveNet	8200	8201	8202	8203
6453	Teleglobe	8210	8211	8212	8213
7018	AT&T (US)	8220	8221	8222	8223
7738	Telemar	8290	8291	8292	8293

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Um cliente com o AS 4111 envia um prefixo com a community "3549:8011 3549:8033 3549:8190"

- O prefixo com essa marca será anunciado da seguinte forma para os peers GLBX:

Qwest (AS 209) receberá o prefixo com o AS PATH: "3549 3549 4111"
MCI (AS 701) receberá o prefixo com o AS PATH: "3549 3549 3549 3549 4111"
OpenTransit (AS 5511) não receberá o prefixo
todos os outros receberão o prefixo normalmente: "3549 4111"

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Intelig

17379:23004 - Community NO-EXPORT-INTERNACIONAL - Não anuncia para peers Internacionais.
17379:23005 - Community NO-EXPORT-NACIONAL - Não anuncia para peers Nacionais.
No-export ? A rota não será anunciada para nenhum peer eBGP da Intelig.
17379:60 = set local-preference 60
17379:70 = set local-preference 70
17379:80 = set local-preference 80
17379:90 = set local-preference 90
Todo anuncio feito, sem a aplicação de communities tem por default o valor 100 de local Preference.

6.4 - Manipulando trânsito (nacional/internacional) com communities

- LAB 1: Considerando as políticas de community para o AS 200:

200:70 = set local-preference 70

200:80 = set local-preference 80

200:90 = set local-preference 90

200:110 = set local-preference 110

- A partir dos AS's clientes, utilizar as communities 200:xxx para alterar o local preference dentro do AS 200, em seguida observar que quando o mesmo é forçado a mudar o local preference para menos de 100, automaticamente começa a ”dar a volta” pelo AS 99, pois acaba escolhendo rotas do AS 99 (local preference 100 padrao)

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Solução:

```
AS-14X-R1#conf t
AS-14X-R1(config)#route-map AS-X-OUT permit 10
AS-14X-R1(config-route-map)#set community none
!--- remove o ”set community” do lab anterior
AS-14X-R1(config-route-map)#set community 200:90
AS-14X-R1(config-route-map)#end
AS-14X-R1#clear ip bgp * soft
```

```
AS-15X-R1#conf t
AS-15X-R1(config)#route-map AS-X-OUT permit 10
AS-15X-R1(config-route-map)#set community 200:70
AS-15X-R1(config-route-map)#set community none
!--- remove o ”set community” do lab anterior
AS-15X-R1(config-route-map)#end
AS-15X-R1#clear ip bgp * soft
```

6.4 - Manipulando trânsito (nacional/internacional) com communities

- LAB 2: De acordo com as políticas do AS 100 abaixo, utilize as communities apropriadas (100:xxxx) para fazer todas as ações:

- Fazer com que o grupo 9 alcance o AS 14**X** pelo AS 200
- Fazer com que o grupo 1 alcance o AS 15**X** pelo AS 200
- Fazer com que o AS 99 alcance os dois (14**X** e 15**X**) pelo AS 200

ASN	Peer	Nao Anuncia	Prepend +1	Prepend +2	Prepend +3
1	Grupo1	1010	1011	1012	1013
2	Grupo2	1020	1021	1022	1023
3	Grupo3	1030	1031	1032	1033
4	Grupo4	1040	1041	1042	1043
5	Grupo5	1050	1051	1052	1053
6	Grupo6	1060	1061	1062	1063
7	Grupo7	1070	1071	1072	1073
8	Grupo8	1080	1081	1082	1083
9	Grupo9	1090	1091	1092	1093
99	Internacional	9910	9911	9912	9913

6.4 - Manipulando trânsito (nacional/internacional) com communities

- **Desafio:**

Grupos de ASN par:

Utilizar as communities da tabela para fazer o AS 100 anunciar seu prefixo com prepend 3 para os ASN's ímpares e prepend 2 para o AS 99 e anunciar normalmente para todo o resto

Grupos de ASN ímpar:

Utilizar as communities da tabela para fazer o AS 100 anunciar seu prefixo com prepend 3 para os ASN's pares e prepend 2 para o AS 99 e anunciar normalmente para todo o resto

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Solução Lab 2:

```
!--- apagando o route map do lab anterior
AS-14X-R1(config)#no route-map AS-X-OUT

AS-14X-R1#conf t
AS-14X-R1(config)#route-map AS-X-OUT permit 10
AS-14X-R1(config-route-map)#set community 100:1090 100:9912
AS-14X-R1(config-route-map)#end
AS-14X-R1#clear ip bgp * soft


!--- apagando o RM do lab anterior
AS-15X-R1(config)#no route-map AS-X-OUT

AS-15X-R1#conf t
AS-15X-R1(config)#route-map AS-X-OUT permit 10
AS-15X-R1(config-route-map)#set community 100:1010 100:9912
AS-15X-R1(config-route-map)#end
AS-15X-R1#clear ip bgp * soft
```

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Solução Lab Desafio:(AS Ímpar)

```
!--- apagando o ROUTE-MAP do lab anterior
AS-14X-R1(config)#no route-map AS-X-OUT

AS-14X-R1#conf t
AS-14X-R1(config)#route-map AS-X-OUT permit 10
AS-14X-R1(config-route-map)#set community 100:9912 100:1023 100:1043
100:1063 100:1083
AS-14X-R1(config-route-map)#end
AS-14X-R1#clear ip bgp * soft

!--- apagando o ROUTE-MAP do lab anterior
AS-15X-R1(config)#no route-map AS-X-OUT

AS-15X-R1#conf t
AS-15X-R1(config)#route-map AS-X-OUT permit 10
AS-15X-R1(config-route-map)#set community 100:9912 100:1013 100:1033
100:1053 100:1073 100:1093
AS-15X-R1(config-route-map)#end
AS-15X-R1#clear ip bgp * soft
```

6.4 - Manipulando trânsito (nacional/internacional) com communities

- Solução Lab Desafio: (AS Par)

```
!--- apagando o RM do lab anterior
AS-14X-R1(config)#no route-map AS-X-OUT

AS-14X-R1#conf t
AS-14X-R1(config)#route-map AS-X-OUT permit 10
AS-14X-R1(config-route-map)#set community 100:9912 100:1013 100:1033 100:1053
100:1073 100:1093
AS-14X-R1(config-route-map)#end
AS-14X-R1#clear ip bgp * soft

!--- apagando o RM do lab anterior
AS-15X-R1(config)#no route-map AS-X-OUT

AS-15X-R1#conf t
AS-15X-R1(config)#route-map AS-X-OUT permit 10
AS-15X-R1(config-route-map)#set community 100:9912 100:1013 100:1033 100:1053
100:1073 100:1093
AS-15X-R1(config-route-map)#end
AS-15X-R1#clear ip bgp * soft
```

6.5
IDENTIFICANDO E TRATANDO ROTAS DE PEERS ESPECÍFICOS

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

6.5 - Identificando e tratando rotas de peers específicos

- Quando utilizamos o comando:

```
AS-14X-R1(config-route-map)#set community xxx:xxx
```

- Estamos substituindo todas as communities que haviam nesse prefixo
- Muitas vezes, é interessante adicionar mais de uma community em "pontos diferentes"

  Ex: Recebo prefixos dos meus downstreams e quero inserir uma community 123:123 para controle interno, utilizando o meio convencional, eu acabaria substituindo uma eventual community que meu cliente utilizou para interagir com um outro AS externo.

- O que fazer?

6.5 - Identificando e tratando rotas de peers específicos

- Com o comando "additive" qualquer community existente é preservada

  AS-**X**-R4(config-route-map)#set community 123:123 additive

- *OBS: Digitar o comando "set community" tem efeito "cumulativo" exemplo: se digitarmos a seguinte sequência de comandos:*

```
router#conf t
router(config)#route-map teste permit 10
router(config-route-map)#set community 123:123
router(config-route-map)#exit
router(config)#route-map teste permit 10
router(config-route-map)#set community 111:222
router(config-route-map)#set community 333:214
router(config-route-map)#end
router#
router#conf t
router(config)#route-map teste permit 10
router(config-route-map)#set community 555:82
```

6.5 - Identificando e tratando rotas de peers específicos

- Todas as communities digitadas serão **ACUMULADAS**:

```
router#sh running-config | begin route-map teste
route-map teste permit 10
 set community 111:222 123:123 333:214 555:82
!
...
```

- Lembrem que o "additive" não tem a ver com esse fato, mas sim com fazer com que as communities inseridas com "additive" sejam "somadas" às communities que já estavam taggeadas no(s) prefixo(s)
- Sem o comando "additive", qualquer community existente é substituida
- Sempre utilizem ”additive” na hora de colocar communities em rotas recebidas de DOWNSTREAMS

7.1
AS DE TRÂNSITO NO PTT

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

7.1 - AS de trânsito no PTT

- Vamos iniciar estabelecendo as sessões BGP de R1 com o PTT-RS1, PTT-RS2 e PTT-LG, em seguida repetir o processo nos DOWNSTREAMS AS-14**X**-R1 e AS-15**X**-R1

- AS-**X**-R1


```
AS-X-R1#conf t
AS-X-R1(config)#router bgp X
!--- como os 2 neighbors terão as mesmas configurações, criaremos um peer-group
AS-X-R1(config-router)#neighbor PTT peer-group
AS-X-R1(config-router)#neighbor PTT remote-as 555
AS-X-R1(config-router)#neighbor PTT soft-reconfiguration inbound
AS-X-R1(config-router)#neighbor PTT route-map PTT-IN in
AS-X-R1(config-router)#neighbor PTT route-map PTT-OUT out
AS-X-R1(config-router)#neighbor PTT send-community both
!--- não precisamos repetir as configurações e apena declarar o peer-group para os dois neighbors
AS-X-R1(config-router)#neighbor 187.16.216.253 peer-group PTT
AS-X-R1(config-router)#neighbor 187.16.216.253 description PTT RS 1
!--- ptt-rs1
AS-X-R1(config-router)#neighbor 187.16.216.254 peer-group PTT
AS-X-R1(config-router)#neighbor 187.16.216.254 description PTT RS 2
!--- ptt-rs2
AS-X-R1(config-router)#neighbor 187.16.216.252 remote-as 555
AS-X-R1(config-router)#neighbor 187.16.216.252 description LG do PTT
!--- o Looking Glass do PTT não fará parte do peer-group pois não defverá receber nenhum filtro in-out
!--- continua no próximo slide...
```

7.1 - AS de trânsito no PTT

- ...BGP com PTT...
- AS-**X**-R1

```
!--- NOTA é preciso levantar o IP de participante na interface
!--- conectada ao PTT
AS-X-R1(config)#interface fastEthernet 0/1
AS-X-R1(config-if)#ip address 187.16.216.X 255.255.255.0
AS-X-R1(config-if)#no shutdown
```


```
!--- continuando...
!--- vamos declarar em networks os prefixos mais específicos
AS-X-R1(config-router)#network X.0.0.0 mask 255.128.0.0
AS-X-R1(config-router)#network X.128.0.0 mask 255.128.0.0
AS-X-R1(config-router)#exit
AS-X-R1(config)#ip route X.0.0.0 255.128.0.0 null 0
AS-X-R1(config)#ip route X.128.0.0 255.128.0.0 null 0
!--- depois garantimos os dois blocos específicos (/9) na FIB
AS-X-R1(config)#end
AS-X-R1#clear ip bgp peer-group PTT out
```

7.1 - AS de trânsito no PTT

- Configurando as políticas de route-map
- AS-**X**-R1

!--- ATENÇÃO TODAS AS PREFIX-LIST ABAIXO JÁ FORAM CRIADAS NOS LABS ANTERIORES
!--- ESTÃO AQUI APENAS PARA LEMBRAR O SEU CONTEÚDO
!--- AS-**X**-R1(config)#ip prefix-list MEUS-BLOCOS permit **X**.0.0.0/8
!--- AS-**X**-R1(config)#ip prefix-list BLOCOS-AS-14**X** permit 14**X**.0.0.0/8
!--- AS-**X**-R1(config)#ip prefix-list BLOCOS-AS-15**X** permit 15**X**.0.0.0/8
!--- CASO ESTAS NÃO ESTEJAM PRESENTES CONFIGUREM NOVAMENTE
!--- vamos configurar primeiro o route-map que trata os filtros de input
AS-**X**-R1(config)#route-map PTT-IN permit 10
AS-**X**-R1(config-route-map)#set local-preference 110
!--- o local preference 110 fará com que todas os routers do AS prefiram as rotas recebidas pelo PTT
!--- em seguida vamos aos route-maps responsáveis pelo output (anúncios) para o PTT
AS-**X**-R1(config)#route-map PTT-OUT permit 10
AS-**X**-R1(config-route-map)#match ip address prefix-list MEUS-BLOCOS
*!--- libera os blocos do AS local **X***
AS-**X**-R1(config-route-map)#exit
AS-**X**-R1(config)#route-map PTT-OUT permit 20
AS-**X**-R1(config-route-map)#match ip address prefix-list BLOCOS-AS-14**X**
!--- libera os blocos do DOWNSTREAM 1
AS-**X**-R1(config-route-map)#exit
AS-**X**-R1(config)#route-map PTT-OUT permit 30
AS-**X**-R1(config-route-map)#match ip address prefix-list BLOCOS-AS-15**X**
!--- libera os blocos do DOWNSTREAM 2
AS-**X**-R1(config-route-map)#end
AS-**X**-R1#clear ip bgp * soft

7.1 - AS de trânsito no PTT

- Todos esses route-maps podem deixar a configuração um pouco maior, porém vai permitir que os ”donos dos AS's” sejam facilmente identificados no ”show run” além de tornar muito fácil de escalar
- Imagine um dos clientes de trânsito AS 14**X** passar a prover trânsito IP para outro AS:

ASN 88 CIDR 88.0.0.0/8

- Com apenas 1 comando o AS 88 está liberado nos filtros de Saída

AS-**X**-R1(config)#ip prefix-list BLOCOS-AS-14**X** permit 88.0.0.0/8

AS-**X**-R1**#show running-config | begin BLOCOS-AS-14X**
ip prefix-list BLOCOS-AS-11x seq 5 permit 14X.0.0.0/8
ip prefix-list BLOCOS-AS-11x seq 10 permit 88.0.0.0/8
!

7.1 - AS de trânsito no PTT

- PS: nos LABs anteriores já configuramos as mesmas políticas para o output do AS 100, vamos apenas conferir:

AS-**X**-R1#**show running-config | begin ANUNCIAR-AS-100**
route-map ANUNCIAR-AS-100 permit 10
match ip address prefix-list MEUS-BLOCOS
!
route-map ANUNCIAR-AS-100 permit 20
*match ip address prefix-list BLOCOS-AS-14**X***
!
route-map ANUNCIAR-AS-100 permit 30
*match ip address prefix-list BLOCOS-AS-15**X***
!

- Como podemos ver na saida do comando, as políticas de output com o AS 100 já estão configuradas

7.1 - AS de trânsito no PTT

- A configuração nos DOWNSTREAMS é muito mais simples:
- AS-14**X**-R1

```
!--- NOTA é preciso levantar o IP de participante na interface
!--- conectada ao PTT
AS-14X-R1(config)#interface fastEthernet 0/1
AS-14X-R1(config-if)#ip address 187.16.216.14X 255.255.255.0
AS-14X-R1(config-if)#no shutdown
```

```
AS-14X-R1#conf t
AS-14X-R1(config)#ip as-path access-list 1 permit ^$
AS-14X-R1(config)#ip route 14X.0.0.0 255.128.0.0 null 0
AS-14X-R1(config)#ip route 14X.128.0.0 255.128.0.0 null 0
AS-14X-R1(config)#router bgp 14X
AS-14X-R1(config-router)#network 14X.0.0.0 mask 255.128.0.0
AS-14X-R1(config-router)#network 14X.128.0.0 mask 255.128.0.0
AS-14X-R1(config-router)#neighbor PTT peer-group
AS-14X-R1(config-router)#neighbor PTT remote-as 555
AS-14X-R1(config-router)#neighbor PTT soft-reconfiguration inbound
AS-14X-R1(config-router)#neighbor PTT filter-list 1 out
AS-14X-R1(config-router)#neighbor PTT weight 50
AS-14X-R1(config-router)#neighbor PTT send-community both
AS-14X-R1(config-router)#neighbor 187.16.216.253 peer-group PTT
AS-14X-R1(config-router)#neighbor 187.16.216.253 description PTT RS 1
AS-14X-R1(config-router)#neighbor 187.16.216.254 peer-group PTT
AS-14X-R1(config-router)#neighbor 187.16.216.254 description PTT RS 2
AS-14X-R1(config-router)#neighbor 187.16.216.252 remote-as 555
AS-14X-R1(config-router)#neighbor 187.16.216.252 description LG do PTT
AS-14X-R1(config-router)#end
```

7.1 - AS de trânsito no PTT

- Seguimos a mesma lógica para o segundo DOWNSTREAM
- AS-15**X**-R1

```
!--- NOTA é preciso levantar o IP de participante na interface
!--- conectada ao PTT
AS-15X-R1(config)#interface fastEthernet 0/1
AS-15X-R1(config-if)#ip address 187.16.216.15X 255.255.255.0
AS-15X-R1(config-if)#no shutdown
```

```
AS-15X-R1#conf t
AS-15X-R1(config)#ip as-path access-list 1 permit ^$
AS-15X-R1(config)#ip route 15X.0.0.0 255.128.0.0 null 0
AS-15X-R1(config)#ip route 15X.128.0.0 255.128.0.0 null 0
AS-15X-R1(config)#router bgp 15X
AS-15X-R1(config-router)#network 15X.0.0.0 mask 255.128.0.0
AS-15X-R1(config-router)#network 15X.128.0.0 mask 255.128.0.0
AS-15X-R1(config-router)#neighbor PTT peer-group
AS-15X-R1(config-router)#neighbor PTT remote-as 555
AS-15X-R1(config-router)#neighbor PTT soft-reconfiguration inbound
AS-15X-R1(config-router)#neighbor PTT filter-list 1 out
AS-15X-R1(config-router)#neighbor PTT weight 50
AS-15X-R1(config-router)#neighbor PTT send-community both
AS-15X-R1(config-router)#neighbor 187.16.216.253 peer-group PTT
AS-15X-R1(config-router)#neighbor 187.16.216.253 description PTT RS 1
AS-15X-R1(config-router)#neighbor 187.16.216.254 peer-group PTT
AS-15X-R1(config-router)#neighbor 187.16.216.254 description PTT RS 2
AS-15X-R1(config-router)#neighbor 187.16.216.252 remote-as 555
AS-15X-R1(config-router)#neighbor 187.16.216.252 description LG do PTT
AS-15X-R1(config-router)#end
```

7.1 - AS de trânsito no PTT

- Vamos entrar no AS-100-R1 (100.100.100.1) e fazer um traceroute para os DOWNSTREAMS do grupo

- Notem que ao entrar no AS do GRUPO, os pacotes são encaminhados para o PTT, e não chegam ao DOWNSTREAM pelo link de trânsito

- Mesmo com um local preference maior, os prefixos que os AS's 14**X** e AS 15**X** anunciam para PTT são mais específicos do que os que anunciam para o trânsito, isso obriga o router R1 a mandar os pacotes pelo PTT (download dos DOWNSTREAMS)

7.1 - AS de trânsito no PTT

- Isso é um problema sério!
- o cliente pode tirar proveito disso e causar prejuízos ”desviando” trânsito IP, obrigando AS local a ”trocar” o trânsito IP por ”nada”
- Uma troca de tráfego onde apenas um tem vantagem...

7.1 - AS de trânsito no PTT

- Para resolver isso, precisamos utilizar uma expressão regular proibindo R1 de receber qualquer anúncio de participantes que também sejam clientes de trânsito do AS


```
AS-X-R1#conf t
AS-X-R1(config)#router bgp X
AS-X-R1(config-router)#neighbor PTT filter-list 50 in
AS-X-R1(config-router)#exit
AS-X-R1(config)#ip as-path access-list 55 deny (^14X_|_14X_|_14X$|^14X$)
AS-X-R1(config)#ip as-path access-list 55 deny (^15X_|_15X_|_15X$|^15X$)
AS-X-R1(config)#ip as-path access-list 55 permit .*
```

- Inserimos o AS de cada cliente de trânsito nessa lista

7.1 - AS de trânsito no PTT

- Vamos efetuar novamente o traceroute partindo de AS-100-R1...

7.2
MANIPULANDO ROTAS DE MÚLTIPLOS PTT'S

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

7.2 - Manipulando rotas de múltiplos PTTs

- Em alguns casos clientes de trânsito solicitam o recebimento apenas de rotas do PTT-SP por exemplo
- Como filtrar e enviar um anúncio assim?
- Uma maneira simples é marcar uma community específica para rotas recebidas:

```
router-SP#conf t
router-SP(config)#route-map PTT-SAOPAULO-IN permit 1
router-SP(config-route-map)#set community 28135:1111 additive
...
```

- Esse comando adiciona a community **28135:1111** e preserva todas que existirem

7.2 - Manipulando rotas de múltiplos PTTs

- Se meu AS participa de mais de um PTT posso filtrar rotas de ambos e exportar para o meu cliente:

```
router-CampinaGrande#conf t
router-CampinaGrande(config)#route-map PTT-CAMPINAGRANDE-IN permit 1
router-CampinaGrande(config-route-map)#set community 28135:2222 additive
```

- Dado o cenário, vamos ver quais seriam os filtros que enviariam para o cliente apenas rotas recebidas do PTT-Campina Grande e PTT-São Paulo:

```
Router(config)#ip community-list 70 permit 28135:1111
Router(config)#ip community-list 70 permit 28135:2222
!
router(config)#route-map CLIENTE-OUT permit 10
router(config-route-map)#match community 70
```

7.3
DOWNSTREAMS COM MULTIHOMING

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

7.3 - DOWNSTREAMS com multihoming

- Os filtros baseados apenas em prefixos nos trazem um problema:

- Como AS de trânsito, geralmente temos vários peers, e caso um dos nossos clientes também tenha outros peers, um filtro baseado unicamente em prefixos pode acabar fazendo o meu AS exportar prefixos errados

- Por exemplo, sou o AS 100, meu cliente é o AS 20, que por sua vez compra trânsito com o AS 200

7.3 - DOWNSTREAMS com multihoming

- Caso o AS 20 anunciasse um bloco específico para cada AS (100 e 200), essa seria minha FIB:

```
router# sh ip bgp 20.0.0.0/20 longer-prefixes
...
   Network          Next Hop            Metric LocPrf Weight Path
*>i20.0.8.0/21      200.100.1.1                           0 200 20 i
*>i20.0.0.0/21      100.20.1.2               300          0 20 i
*>i20.0.0.0/20      100.20.1.2               300          0 20 i
*                   200.100.1.1                           0 200 20 i
...
```

- Percebam que por ser válido na FIB, o prefixo 20.0.8.0/21 seria exportado para todos os peers externos (menos para o AS 200 de onde "aprendemos") começando pelo AS 100

7.3 - DOWNSTREAMS com multihoming

- Precisamos adicionar mais uma condição de match além do prefix list com expressões regulares
- Esse é o route-map padrão que já conhecemos:

```
router(config)#ip prefix-list BLOCOS-AS-20-ACL33 permit 20.0.0.0/8 le 24

router(config)#route-map ANUNCIAR-AS-XXX permit 10
router(config-route-map)#match ip address prefix-list BLOCOS-AS-20-ACL33

!--- Adicionamos a seguinte acl:
router(config)#ip as-path access-list 33 permit ^(20_|20$)
!--- apenas se a rota tiver sido ORIGINADA pelo AS 20

!--- voltamos no mesmo route-map
router(config)#route-map ANUNCIAR-AS-XXX permit 10
!--- e adicionamos a condição
router(config-route-map)#match as-path 33
```

7.3 - DOWNSTREAMS com multihoming

- A nomeclatura da prefix-list com final (ACL-33) é para facilitar, pois indica o número da ACL de as-path correspondente ao cliente
- Nos routers Cisco não há possibilidade de nomear uma ACL de as-path, podemos apenas utilizar apenas um número
- Para quem usa Quagga é possível atribuir um nome para a ACL de as-path e community lists o que deixa tudo ainda mais fácil:

```
router-quagga(config)#ip prefix-list BLOCOS-AS-20 permit 20.0.0.0/8 le 24
router-quagga(config)#ip as-path access-list PATH-AS-20 permit ^(20_|20$)

router-quagga(config)#route-map ANUNCIAR-AS-XXX permit 10
router-quagga(config-route-map)#match ip address prefix-list BLOCOS-AS-20
router-quagga(config-route-map)#match as-path PATH-AS-20
```

7.4
Clientes do mesmo AS em POPs distintos

v1.1 | Autor: Rinaldo Vaz | rinaldopvaz@gmail.com

7.4 - Clientes do mesmo AS em POPs distintos

- Os filtros que aprendemos em 7.3 ainda nos trazem um problema caso um cliente do mesmo AS compre link em 2 POPs distintos

- Devido à rota do POP A ser refletida pelo RR para o POP B, não haveria alterações no AS Path

- Os prefixos que deveriam ter sido anunciados ao UPSTREAM local do POP A também acabam sendo anunciados no POP B.

7.4 - Clientes do mesmo AS em POPs distintos

- Para resolver esse problema, precisamos fazer com que o RR insira uma community "Local-AS"

```
route-reflector(config)#route-map ANUNCIAR-POR-IBGP permit
10
route-reflector(config-route-map)#set community local-AS additive
```

- Dessa maneira eliminamos esse risco, porém qualquer POP interligado em um PTT acabaria não anunciando essas redes, e isso é ruim

7.4 - Clientes do mesmo AS em POPs distintos

- Uma forma de resolver isso é utilizando outra community no RR, exemplo:

```
route-reflector(config)#route-map ANUNCIAR-POR-IBGP permit 10
route-reflector(config-route-map)#set community 1:1 additive
```

- Nos pops remotos preciso de um RM de deny para todos os UPSTREAMS:

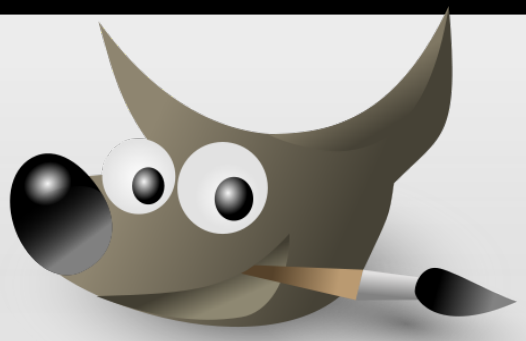
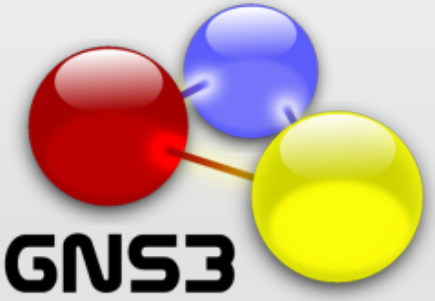
```
Router-PTT-Natal(config)#ip community-list 10 permit 1:1
!--- importante lembrar que o permit da ACL terá função de deny
!--- caso esteja dentro de um RM de "deny" como abaixo

Router-PTT-Natal(config)#route-map OPERADORA-X-OUT deny 1
Router-PTT-Natal(config-route-map)#match community 10
```

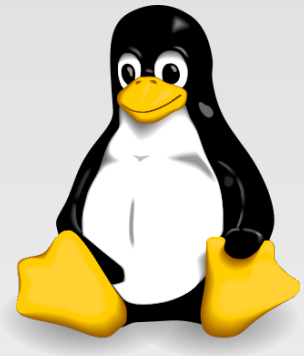
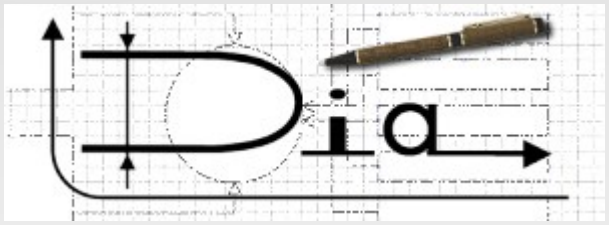
7.4 - Clientes do mesmo AS em POPs distintos

- Para resolver esse problema, precisamos fazer com que o RR insira uma community "Local-AS"

```
route-reflector(config)#route-map ANUNCIAR-POR-IBGP permit 10
route-reflector(config-route-map)#set community local-AS additive
```

- Dessa maneira eliminamos esse risco, porém qualquer POP interligado em um PTT acabaria não anunciando essas redes, e isso é ruim

- 100% desse material foi elaborado utilizando software livre, e é livre para ser copiado e distribuído desde que gratuitamente.
- Está é a versão **1.1**
- Para reportar erros ou solicitar a versão mais recente entre em contado por email: **rinaldopvaz@gmail.com**
- Autor: Rinaldo Vaz

- CHANGELOG versão 1.0 – 1.1:

  23/04/2012

  - **2.6 Slide de abertura do sub-capítulo:**
  - ✔ *corrigido slide 2.6 trocado pelo 2.7*
  - **2.7 Corrigido erro de digitação:**
  - ✘ *erro: "...último /10..."*
  - ✔ *correção: "... último /9..."*
  - **3.1 saída do comando "sh ip bgp neighbors (peering) advertise-routes":**
  - ✘ *corrigidos 2 prefixos que apareciam errados como /10,*
  - ✔ *o correto são 3 prefixos, um /8 e dois /9*
  - **3.1 saída do comando "sh ip bgp regexp ^2$":**
  - ✘ *corrigidos 2 prefixos que apareciam errados como /10,*
  - ✔ *o correto são 3 prefixos, um /8 e dois /9*

- CHANGELOG versão 1.0 – 1.1:

  23/04/2012

  - ***3.5 Descrição de participante remoto no PTT:***
  - ✔ *a explicação ficou mais clara com relação às consequências da latência mais alta*
  - ***4.2 Precauções para filtros de UPSTREAMS:***
  - ✗ *Erro: "... Garantir que rotas de **DOWNSTREAMS** não recebam prioridade maior..."*
  - ✔ *Correção: "...Garantir que rotas de **UPSTREAMS** nao recebam prioridade maior que rotas DOWNSTREAMS..."*
  - ***4.2 Utilização do (ge) e (le), corrigido erro de digitação:***
  - ✗ *Erro: "... aceitar qualquer prefixo **/25** e descartar..."*
  - ✔ *Correção: "... aceitar qualquer prefixo **/24** e descartar..."*

- CHANGELOG versão 1.0 – 1.1:

  23/04/2012

  - ***4.2 adicionando prefixos na lista "NAO-RECEBER", corrigidos acões de permit erradas:***
  - *Erro: ...seq 70* ***permit****...*
  - *...seq 80* ***permit****...*
  - *...seq 90* ***permit****...*
  - *Correção: ...seq 70* ***deny****...*
  - *...seq 80* ***deny****...*
  - *...seq 90* ***deny****...*
  - ***4.6 expressão regular de parcial, modificada para apenas uma linha, e adicionado um terceiro exemplo para parcial de 3 AS's no path:***
  - *nova expressão: ^([0-9]+|[0-9]+ [0-9]+)$*

- CHANGELOG versão 1.0 – 1.1:

  23/04/2012

  - ***4.8 Mudança de título:***
  - ✗ *Anterior: Manipulando downstream com route-map e prepend*
  - ✓ *Atual: Manipulando trânsito dos downstreams com prepend*
  - ***6.1 e 6.2 Sub-capítulo sobre expressões regulares:***
  - ✓ *Movidos de 6.1/6.2 para 4.4/4.5 , 4.4 mudou para 4.6, 4.5 para 4.7, 4.6 para 4.8 e 4.7 para 4.9*
  - ***5.1 Diagrama do cenário 5: Corrigido terceiro octeto da rede dos laptops***
  - ✗ *Erro: x.x.100.0/24*
  - ✓ *Correção: x.x.200.0/24*

- CHANGELOG versão 1.0 – 1.1:

  23/04/2012

  - ***5.1 Diagrama do cenário 5:***

  - *a partir da versão 1.2 Serão modificados número dos AS's clientes de 11x e 12x para 14x e 15x (para o grupo 7 ocorria um erro ao utilizar o IP 127.0.0.1)*

  - ***5.1 configurações iBGP, corrigido IP errado na configuração de R1 com R5***

  - *Erro:"...neighbor X.0.0.2 remote-as X"*

    "...neighbor X.0.0.2 description iBGP Router 2

  - *Correção: "...neighbor X.0.0.5 remote-as X"*

    "...neighbor X.0.0.5 description iBGP Router 5

- CHANGELOG versão 1.0 – 1.1:

  23/04/2012

  - **5.1 configuração eBGP com AS 200 (segunda rodada), corrigido endereço incorreto de neighbor:**

  ✗ *erro: "...neighbor 200.X.1.1 remote-as 100"*

  ”...neighbor 200.X.1.1 soft-reconfiguration inbound"

  "...neighbor 200.X.1.1 route-map ANUNCIAR-AS-200 out"

  ✓ *correção: "...neighbor 200.X.2.1 remote-as 200"*

  "...neighbor 200.X.2.1 soft-reconfiguration inbound"

  "...neighbor 200.X.2.1 route-map ANUNCIAR-AS-200 out"

  - ***5.1 Configuração iBGP entre R2 > R3, corrigido número da loopback faltando***

  ✗ *erro: "...neighbor X.0.0.3 update-source loopback "*

  ✓ *correção: "...neighbor X.0.0.3 update-source loopback 0"*

- CHANGELOG versão 1.0 – 1.1:

  23/04/2012

  - ***5.1 Configuração iBGP entre R3>todos***
  - ✗ *erro: alguns IPs errados*
  - ✓ *correção: corrigidos IPs corretos para R1, R2, R4 e R5*
  - **6.1 LAB 2, communities padrão**
  - ✓ *Incluida uma explicação mais clara dos objetivos do LAB*
  - **7.3 Corrigida expressão regular de permitir saida apenas de prefixos originados ”diretamente” do downstream:**
  - ✗ *erro: ”_20$”*
  - *forma correta: ”(20_|20$)”*

  - ***Adicionadas mais imagens para facilitar a visualização das configurações***