

# UTILIZANDO FILTROS BGP NO MIROTIK

## Objetivo:

Proporcionar um melhor entendimento com relação ao funcionamento da Internet, trazendo um ponto de vista pouco explorado no roteamento entre sistemas autônomos



Apresentação online atualizada:

[https://docs.google.](https://docs.google.com/presentation/d/1vo78vXgt7zQY0Qo1HgZPUuX3wa8Fza9oHBASZz6HDME/edit)

[com/presentation/d/1vo78vXgt7zQY0Qo1HgZPUuX3wa8Fza9oHBASZz6H  
DME/edit](https://docs.google.com/presentation/d/1vo78vXgt7zQY0Qo1HgZPUuX3wa8Fza9oHBASZz6HDME/edit)

Link resumido:

<http://t.co/c8G5jS3e>

# Palestrante:

Rinaldo Vaz

Formado em redes de computador

Trabalha na área de redes desde 2001

Usuário avançado do RouterOS desde 2007

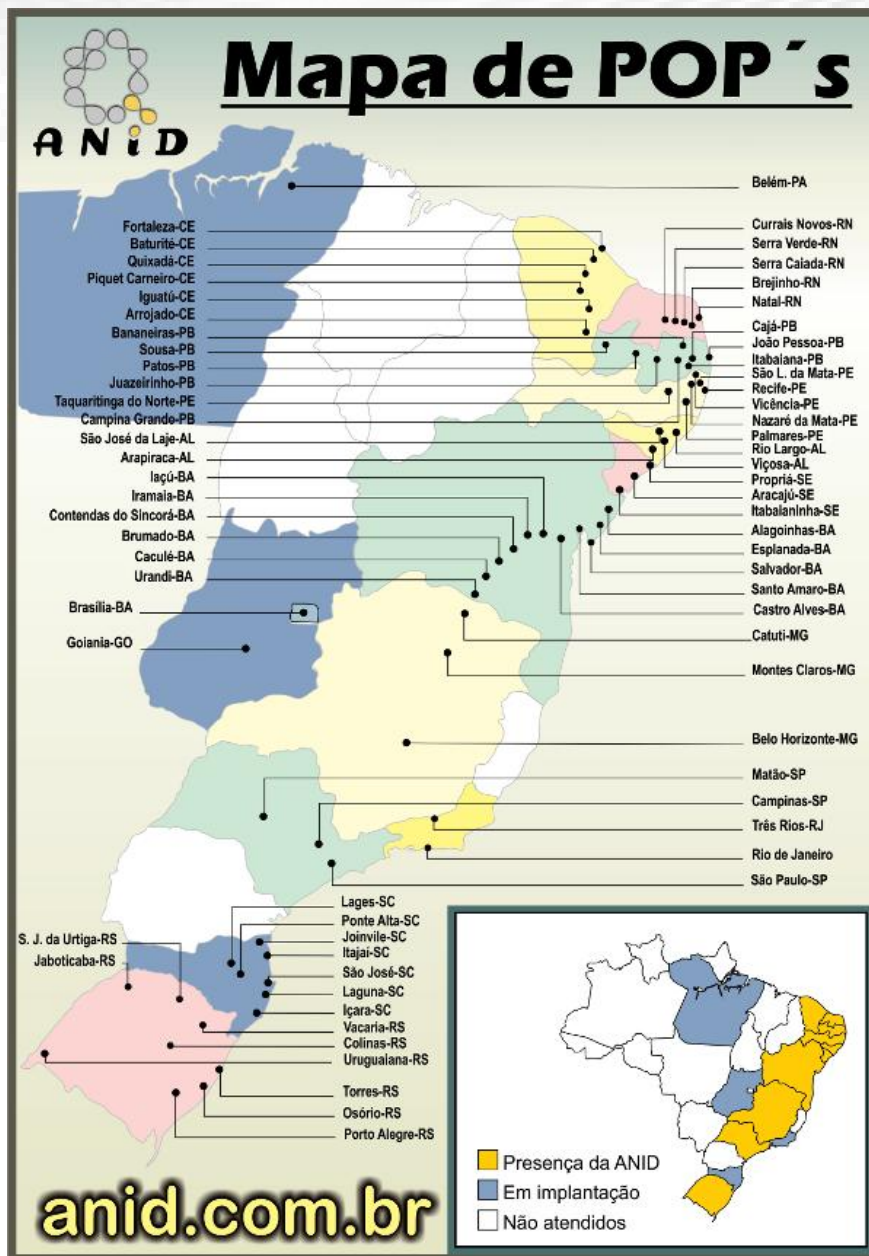
Certificado MTCNA - 2009 / MTCRE - 2010

Analista de Redes da Associação Nacional para  
Inclusão Digital (ANID) desde 2010

Chefe de Operações do NOC

Responsável técnico pelo AS 28135





# **UTILIZANDO FILTROS BGP NO MIKROTIK**

- 1- Contratando mais de uma operadora**
- 2- Anunciando corretamente nas trocas de trafego**
- 3- Boas práticas para um AS de transito**
- 4- Utilizando BGP communities**
- 5- Estudo de Caso**

# **1-Contratando mais de uma operadora**

## **1.1 Balanceamento de links**

\*Antes de começarmos a utilizar BGP precisamos inicialmente abandonar um pré-conceito natural de ver a internet partindo do nosso ponto de vista

Com o BGP, os ANÚNCIOS QUE FAÇO interferem no meu DOWNLOAD, e os ANÚNCIOS QUE RECEBO interferem no meu UPLOAD, portanto se quero que meu UPLOAD vá pela operadora A, preciso convencer o MEU ROTEADOR de que a operadora A é a melhor opção manipulando os seguintes atributos nas rotas BGP:

# 1-Contratando mais de uma operadora

## 1.1 Balanceamento de links

Critério de desempate ao escolher a melhor rota:

maior WEIGHT.

Maior LOCAL PREFERENCE

Rede com origem LOCAL

Menor AS\_PATH.

Menor tipo de ORIGIN

Menor multi-exit discriminator (MED)

Rotas eBGP em relação a rotas iBGP

Mais baixo IGP METRIC.

A rota mais antiga (estável)

Recebidas do peer com menor ROUTER-ID

Menor CLUSTER LIST (ambiente Route Reflector)

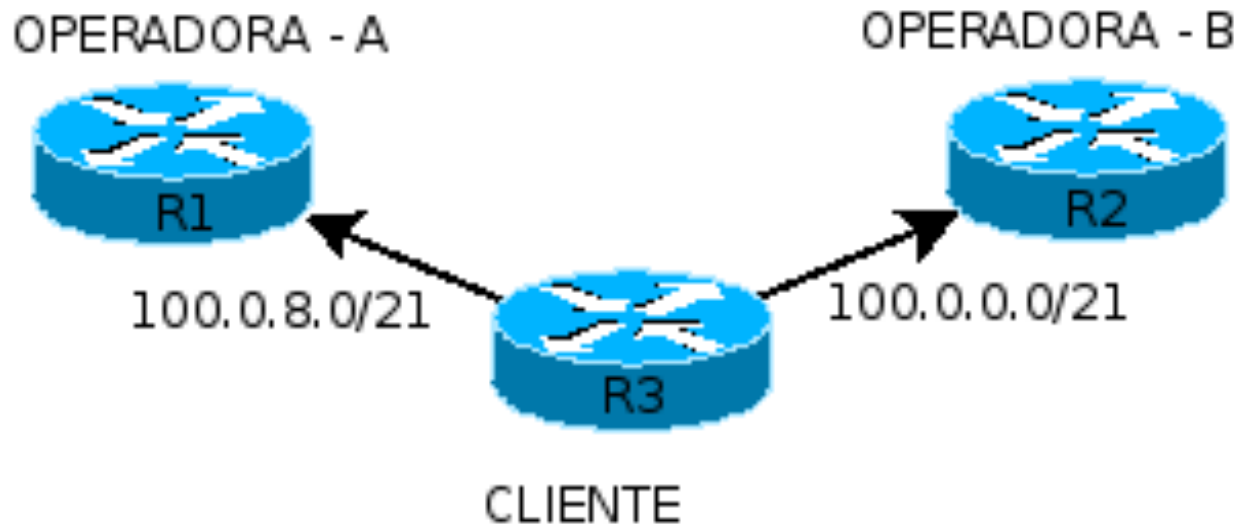
Menor endereço IP de peer

# Porém o primeiro fator a ser analisado é menor bloco, sendo assim, uma rota /24 vai ser considerada sempre a melhor rota em relação a um /23, e todos os atributos BGP serão ignorados

# 1-Contratando mais de uma operadora

## 1.1 Balanceamento de links

Se quero que meu DOWNLOAD venha pela operadora A, preciso convencer os ROTEADORES DA INTERNET que o melhor caminho é pela operadora A, e nesse caso a **maioria**\* dos atributos BGP não vão fazer diferença, já que esses roteadores estão a vários AS's de distância. A maneira mais simples é anunciar metade do meu bloco para operadora A (100.0.8.0/21) e a outra metade para operadora B (100.0.0.0/21)



\*Apenas manipulações de AS-path e communities são preservadas através dos sistemas autônomos

# 1-Contratando mais de uma operadora

## 1.1 Balanceamento de links

Configurando:

```
/routing filter>  
add action=accept chain=operadoraA-out prefix=100.0.8.0/21  
add action=accept chain=operadoraA-out prefix=100.0.0.0/20  
add action=discard chain=operadoraA-out  
  
add action=accept chain=operadoraB-out prefix=100.0.0.0/21  
add action=accept chain=operadoraB-out prefix=100.0.0.0/20  
add action=discard chain=operadoraB-out  
  
/routing bgp network>  
add network=100.0.0.0/21 synchronize=no  
add network=100.0.8.0/21 synchronize=no  
add network=100.0.0.0/20 synchronize=no
```

\*A simetria do balanceamento provavelmente não será alcançada, porém a distribuição equalizada das redes para seus clientes finais possibilita uma boa aproximação. Nos slides seguintes veremos a razão de publicarmos o bloco inteiro para ambas operadoras



# **1-Contratando mais de uma operadora**

## **1.2 FAIL OVER automático com BGP**

Conforme vimos em 1.1, as rotas "mais específicas" serão sempre preferidas pelos roteadores da internet, sendo assim, se eu anunciar também o meu bloco INTEIRO (100.0.0.0/20) para as duas operadoras, essa rota ficará válida em suas tabelas, porém não será utilizada...

...a não ser e claro, que uma das operadoras fique fora!

# 1-Contratando mais de uma operadora

## 1.2 FAIL OVER automático com BGP

Rotas com as 2 operadoras UP

100.0.0.0/21 => as-path 550,650,750,850,950,100 1#

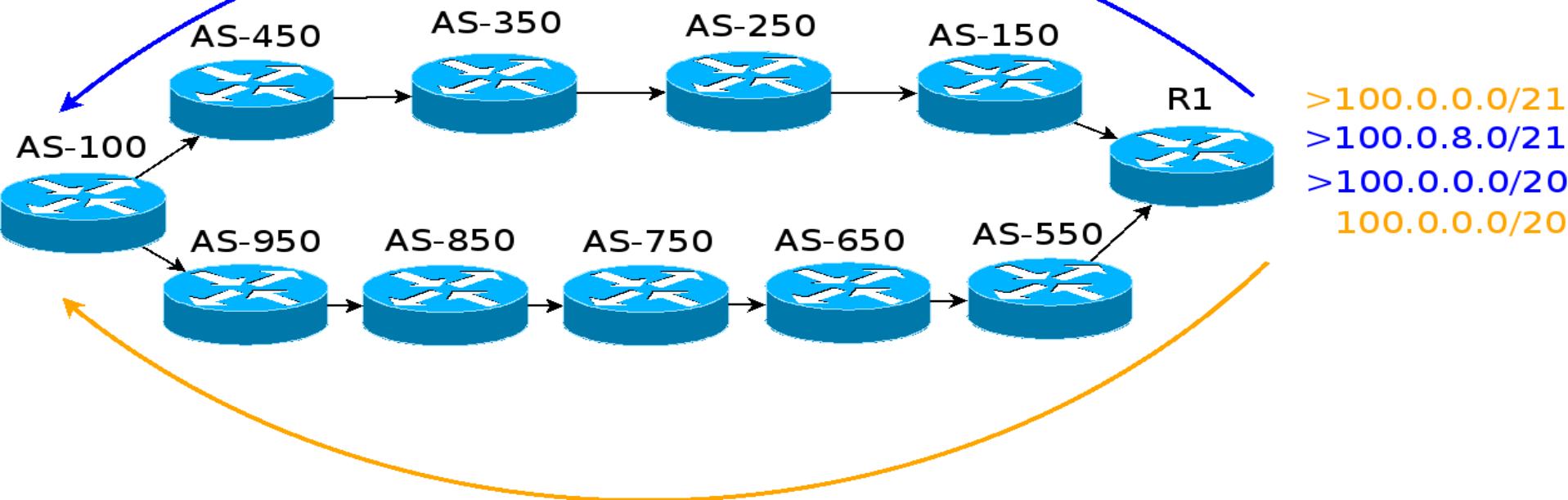
100.0.8.0/21 => as-path 150,250,350,450,100 2#

100.0.0.0/20 => as-path 150,250,350,450,100 3# melhor rota mas nao utilizada

100.0.0.0/20 => as-path 550,650,750,850,950,100 4#

Para alcançar 100.0.8.1 ate 100.0.15.254 o roteador vai escolher a rota #2

Para alcançar 100.0.0.1 ate 100.0.7.254 o roteador vai escolher a rota #1



# 1-Contratando mais de uma operadora

## 1.2 FAIL OVER automático com BGP

```
[admin@R1] > ip route print detail where dst-address in 100.0.0.0/20
```

Flags: X - disabled, A - active, D - dynamic,  
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,  
B - blackhole, U - unreachable, P - prohibit

```
0 ADb dst-address=100.0.0.0/21 gateway=192.168.1.1  
    gateway-status=192.168.1.1 reachable ether1-link-AS_550 distance=20  
    scope=40 target-scope=10 bgp-as-path="550,650,750,850,950,100" bgp-origin=igp  
received-from=peer_AS-550
```

```
1 Db dst-address=100.0.0.0/20 gateway=192.168.1.1  
    gateway-status=192.168.1.1 reachable ether1-link-AS_550 distance=20  
    scope=40 target-scope=10 bgp-as-path="550,650,750,850,950,100" bgp-origin=igp  
received-from=peer_AS-550
```

```
2 ADb dst-address=100.0.8.0/21 gateway=192.168.2.1  
    gateway-status=192.168.2.1 reachable ether2-link-AS_150 distance=20  
    scope=40 target-scope=10 bgp-as-path="150,250,350,450,100" bgp-origin=igp  
received-from=peer_AS-150
```

```
* 3 ADb dst-address=100.0.0.0/20 gateway=192.168.2.1  
    gateway-status=192.168.2.1 reachable ether2-link-AS_150 distance=20  
    scope=40 target-scope=10 bgp-as-path="150,250,350,450,100" bgp-origin=igp  
received-from=peer_AS-150
```

# 1-Contratando mais de uma operadora

## 1.2 FAIL OVER automático com BGP

Rotas com a operadora do as 450 down

100.0.0.0/21 => as-patch 550,650,750,850,950,100 1#

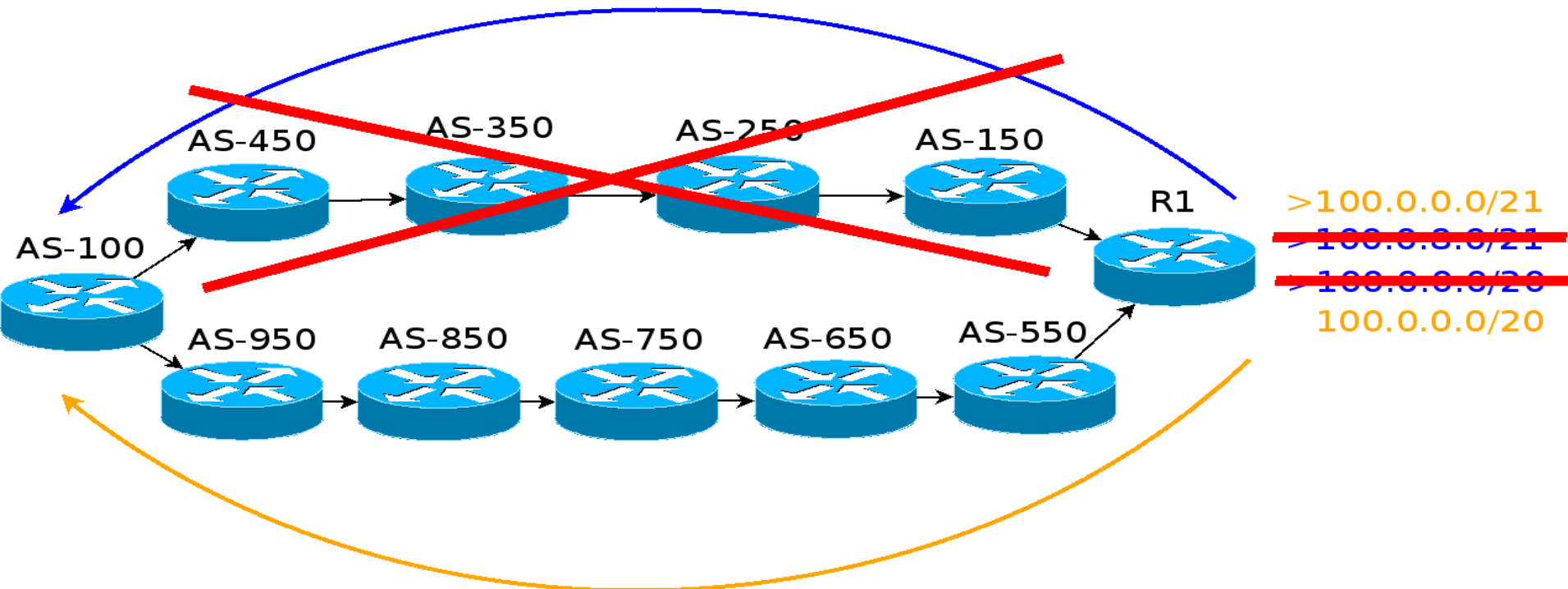
100.0.8.0/21 => as-patch 150,250,350,450,100 2#

removida

100.0.0.0/20 => as-patch 150,250,350,450,100 3#

removida

100.0.0.0/20 => as-patch 550,650,750,850,950,100 4#



# 1-Contratando mais de uma operadora

## 1.2 FAIL OVER automático com BGP

```
[admin@R1] > ip route print detail where dst-address in 100.0.0.0/20
```

Flags: X - disabled, A - active, D - dynamic,  
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,  
B - blackhole, U - unreachable, P - prohibit

```
0 ADb dst-address=100.0.0.0/21 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether1-link-AS_550 distance=20  
scope=40 target-scope=10 bgp-as-path="550,650,750,850,950,100" bgp-origin=igp  
received-from=peer_AS-550
```

```
1 ADb dst-address=100.0.0.0/20 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether1-link-AS_550 distance=20  
scope=40 target-scope=10 bgp-as-path="550,650,750,850,950,100" bgp-origin=igp  
received-from=peer_AS-550
```

```
2 ADb dst-address=100.0.8.0/21 gateway=192.168.2.1  
gateway-status=192.168.2.1 reachable ether2-link-AS_150 distance=20  
scope=40 target-scope=10 bgp-as-path="150,250,350,450,100" bgp-origin=igp received-  
from=peer_AS-150
```

```
3 ADb dst-address=100.0.0.0/20 gateway=192.168.2.1  
gateway-status=192.168.2.1 reachable ether2-link-AS_150 distance=20  
scope=40 target-scope=10 bgp-as-path="150,250,350,450,100" bgp-origin=igp received-  
from=peer_AS-150
```

# **1-Contratando mais de uma operadora**

## **1.3 FAIL OVER automático com BGP - UPLOAD**

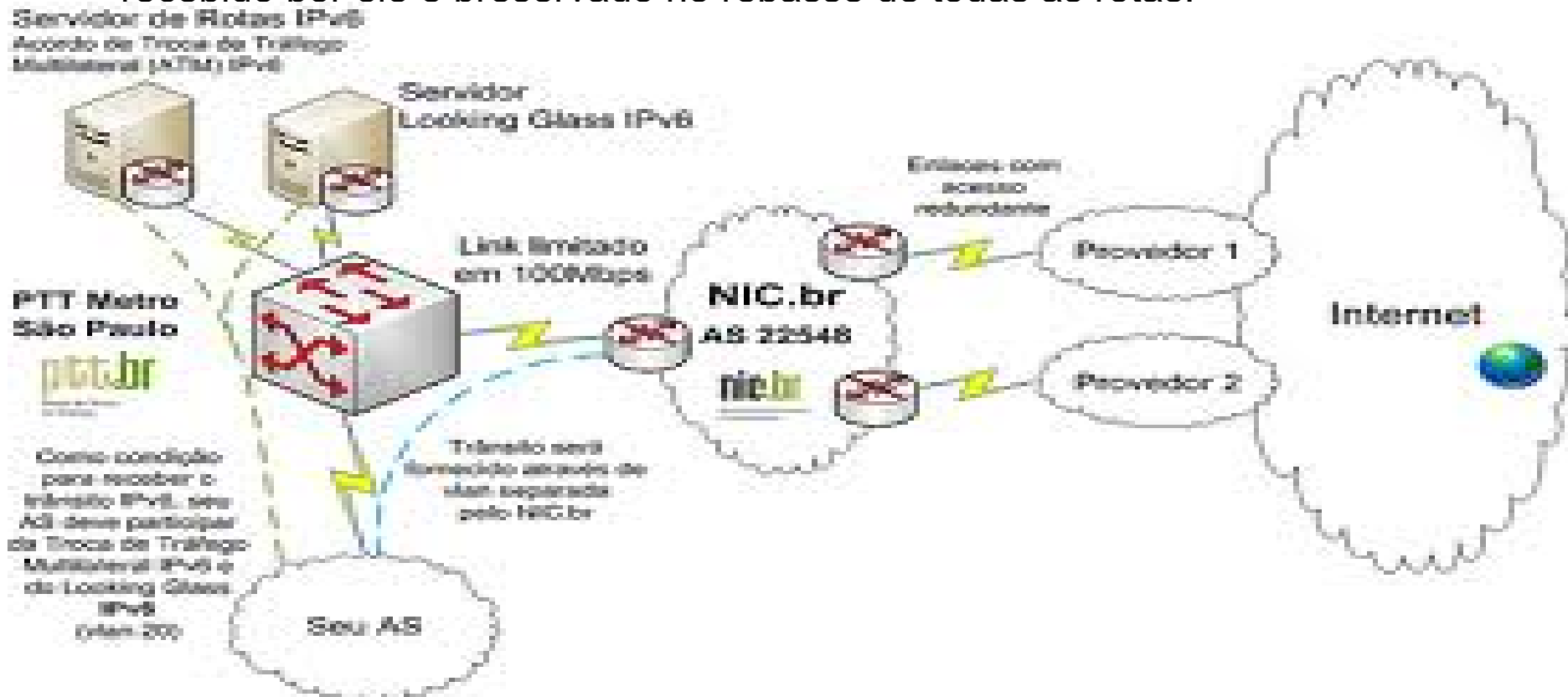
Para que funcione corretamente é preciso receber a tabela full das duas operadoras, pois numa eventual queda não basta o anúncio chegar a internet sem que meu router conheça a rota

\*caso não haja intenção de receber a tabela full, é preciso então receber a "rota default" via BGP das duas operadoras, e garantir que nenhuma "rota default" esteja com distância administrativa menor que 21. (recomenda-se 201 em função da distância 200 das rotas iBGP)

## 2-Participando das trocas de tráfego

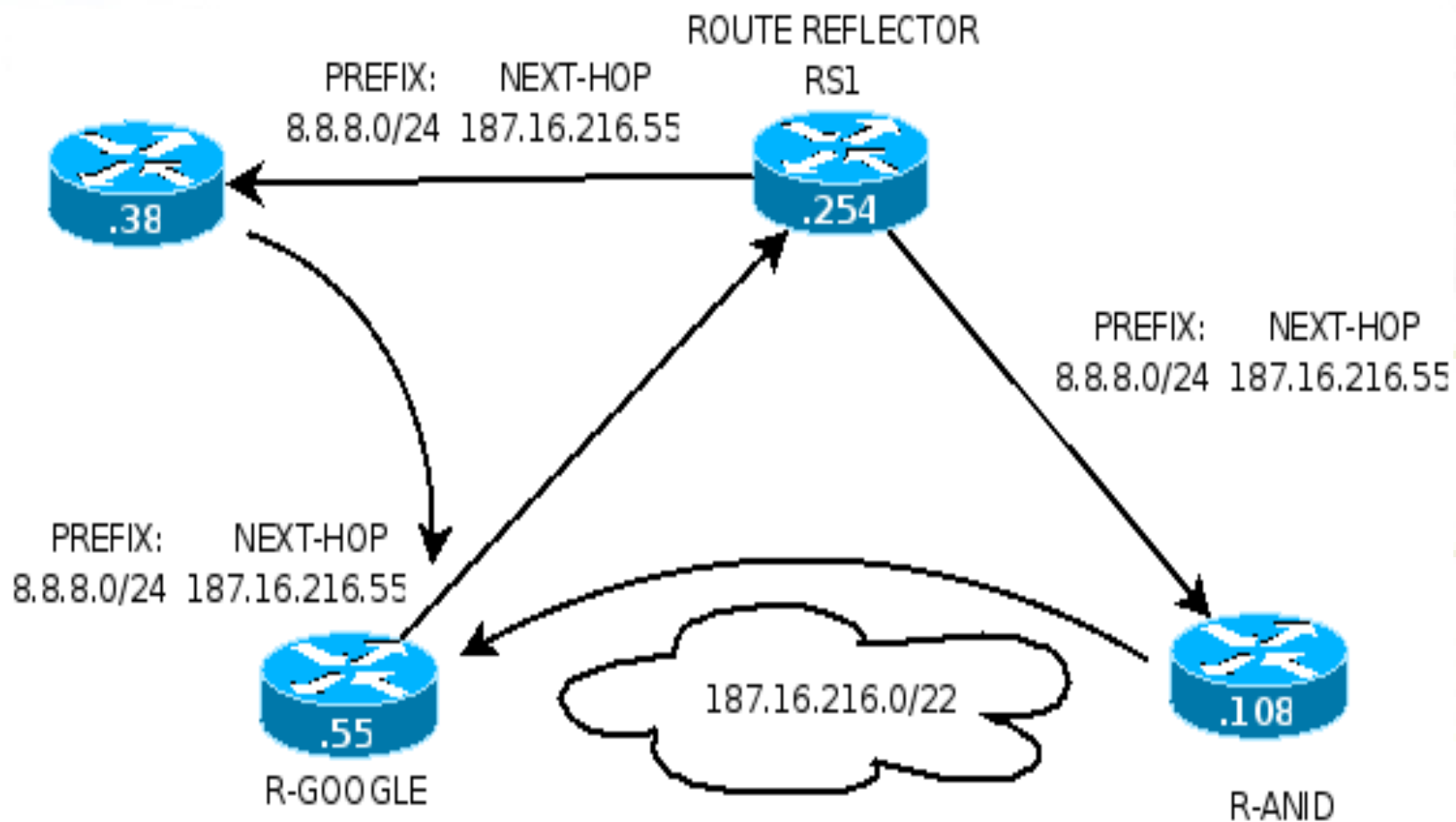
### 2.1 Como funciona uma troca de tráfego

Um ponto de troca de tráfego é um "domínio de broadcast" compartilhado por todos os participantes, que estabelecem conexões BGP no modelo "mídia compartilhada". Apenas um roteador estabelece sessões BGP com todos os outros trabalhando em modo "route-reflector", assim, o next-hop recebido por ele é preservado no repasse de todas as rotas.



## 2-Participando das trocas de tráfego

### 2.2 O que fazer com as rotas recebidas





## **2-Participando das trocas de tráfego**

### **2.2 O que fazer com as rotas recebidas**

Utilizaremos como exemplo o PTT-Metro de São Paulo onde atualmente centenas de AS's participam, totalizando uma media atual de 6000 rotas que serão recebidas por cada peer (rs1 e rs2). Estas rotas são duplicadas para o caso de falha em um dos routers, por isso é importante os 2 peers estarem ativos SEMPRE.

É provável que rotas recebidas dos rs1 e 2 também sejam recebidas da sua operadora, sendo assim, utilizamos os atributos BGP para "forçar prioridade" à essas rotas.

## 2-Participando das trocas de tráfego

### 2.2 O que fazer com as rotas recebidas

Configurando os filtros:

```
/routing filter>
```

```
add chain=ptt-in set-bgp-local-pref=110 action=passthrough
```

\*com action "accept" vai funcionar?

PS: Apenas o atributo Local Preference deveria fazer o RouterOS escolher como a melhor rota, porém nenhum atributo BGP tem funcionado nas versões que testamos na arquitetura PPC, e para isso utilizaremos uma "STA"

```
/routing filter>
```

```
add chain=ptt-in set-distance=19 action=passthrough
```

## #2 minutos de debate

### Alguém já teve esse problema?

A rota #1 deveria ter sido considerada a melhor ao invés da #0?

```
[admin@JPAPBRBSEDE] > ip route print detail where dst-address in 187.86.80.0/20
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
0 ADb  dst-address=187.86.80.0/20 gateway=187.86.0.253
      gateway-status=187.86.0.253 reachable vlan-115-link-anid distance=20 scope=40
      target-scope=10 bgp-as-path="28135,262681" bgp-origin=igp
      bgp-communities=28135:2222 bgp-ext-communities="RT:28135:2222" received-from=bsd
1 Db  dst-address=187.86.80.0/20 gateway=10.110.20.1
      gateway-status=10.110.20.1 reachable vlan-115-link-anid distance=200 scope=40
      target-scope=30 bgp-as-path="262681" bgp-local-pref=900 bgp-origin=igp
      bgp-communities=28135:2222 bgp-ext-communities="RT:28135:2222"
      received-from=as28135
[admin@JPAPBRBSEDE] > 
```

## 2-Participando das trocas de tráfego

### 2.3 Anunciando corretamente

Os anúncios que fazemos para as trocas de tráfego devem logicamente ser interpretados como "melhores caminhos" para os participantes diretos e indiretos, sendo assim, devemos anunciar blocos mais específicos.

Como já anunciei dois /21 para minhas duas operadoras, anunciarei para as trocas 4 blocos /22.

```
/routing bgp network>
```

```
add network=100.0.0.0/20 synchronize=no
```

```
add network=100.0.0.0/21 synchronize=no
```

```
add network=100.0.8.0/21 synchronize=no
```

```
add network=100.0.0.0/22 synchronize=no
```

```
add network=100.0.4.0/22 synchronize=no
```

```
add network=100.0.8.0/22 synchronize=no
```

```
add network=100.0.12.0/22 synchronize=no
```

## 2-Participando das trocas de tráfego

### 2.3 Anunciando corretamente

Existe mais de 1 maneira de fazer os filtros

```
/routing filter>  
add action=accept chain=ptt-out prefix=100.0.0.0/22  
add action=accept chain=ptt-out prefix=100.0.4.0/22  
add action=accept chain=ptt-out prefix=100.0.8.0/22  
add action=accept chain=ptt-out prefix=100.0.12.0/22  
add action=discard chain=ptt-out
```

OU

```
add action=accept chain=ptt-out prefix=100.0.0.0/20 prefix-  
length=22  
add action=discard chain=ptt-out
```

## 2-Participando das trocas de tráfego

### 2.3 Anunciando corretamente

Assim como nós, os roteadores participantes também terão em sua tabela rotas que anunciamos para nossas 2 operadoras

100.0.0.0/21 => as-patch 550,650,750,850,950,100	1#
100.0.8.0/21 => as-patch 150,250,350,450,100	2#
100.0.0.0/20 => as-patch 150,250,350,450,100	3#
100.0.0.0/20 => as-patch 550,650,750,850,950,100	4#
100.0.0.0/22 => as-patch 100	5#
100.0.4.0/22 => as-patch 100	6#
100.0.8.0/22 => as-patch 100	7#
100.0.12.0/22 => as-patch 100	8#

Desse modo, não importa a maneira que o administrador do configure os atributos BGP, os as rotas com /22 serão sempre escolhidas.

## 2-Participando das trocas de tráfego

### 2.4 Perguntas frequentes

Os anuncios que faço para os PTTs não poderiam ser /20-21 já que os participantes terão apenas 1 AS no patch e/ou configuram um local preference maior para rotas recebidas de mim?

*R- Poderiam, mas anunciar blocos /22 garantem que os participantes "indiretos" escolham as rotas anunciadas por mim para o PTT independente das configurações do administrador do AS remoto*

## **3-Boas práticas para um AS de transito**

### **3.1 - Evitando receber dos clientes anúncios indesejados**

Quando precisamos propagar blocos de clientes, é interessante não receber nada que não seja combinado, mas ao mesmo tempo permitir ao cliente uma certa "liberdade" para decidir quais sub-blocos anunciar para você:

Exemlo para os próximos slides:

Cliente:

AS 100

Bloco CIDR 100.0.0.0/20



## 3-Boas práticas para um AS de transito

### 3.1 - Evitando receber dos clientes anúncios indesejados

```
/routing bgp peer>  
add default-originate=if-installed in-filter=as100-in out-filter=as100-out  
name=cliente_as100 nexthop-choice=force-self remote-address=192.  
168.0.2 remote-as=100  
  
routing filter>  
add action=accept chain=as100-in prefix=100.0.0.0/20 prefix-  
length=20-24 set-bgp-local-pref=120  
add action=discard chain=as100-in
```

#traduzinho: quero receber a rede 100.0.0.0/20 ou qualquer sub-bloco /21, /22, /23 ou /24

\*Evitamos com esse filtro que cheguem em nossa tabela anúncios errados que o cliente venha a fazer, por exemplo, se ele desabilita seu filtro de saída, evitamos receber dele a tabela-full (360000 rotas em media), o impacto disso seria todo o meu upload subir pelo cliente, já que as rotas recebidas por ele recebem uma prioridade maior do que as rotas recebidas pelas minhas operadoras.

## **3-Boas práticas para um AS de transito**

### **3.2 - Evitando propagar anúncios do seu cliente, mas que foram feitos para outra operadora.**

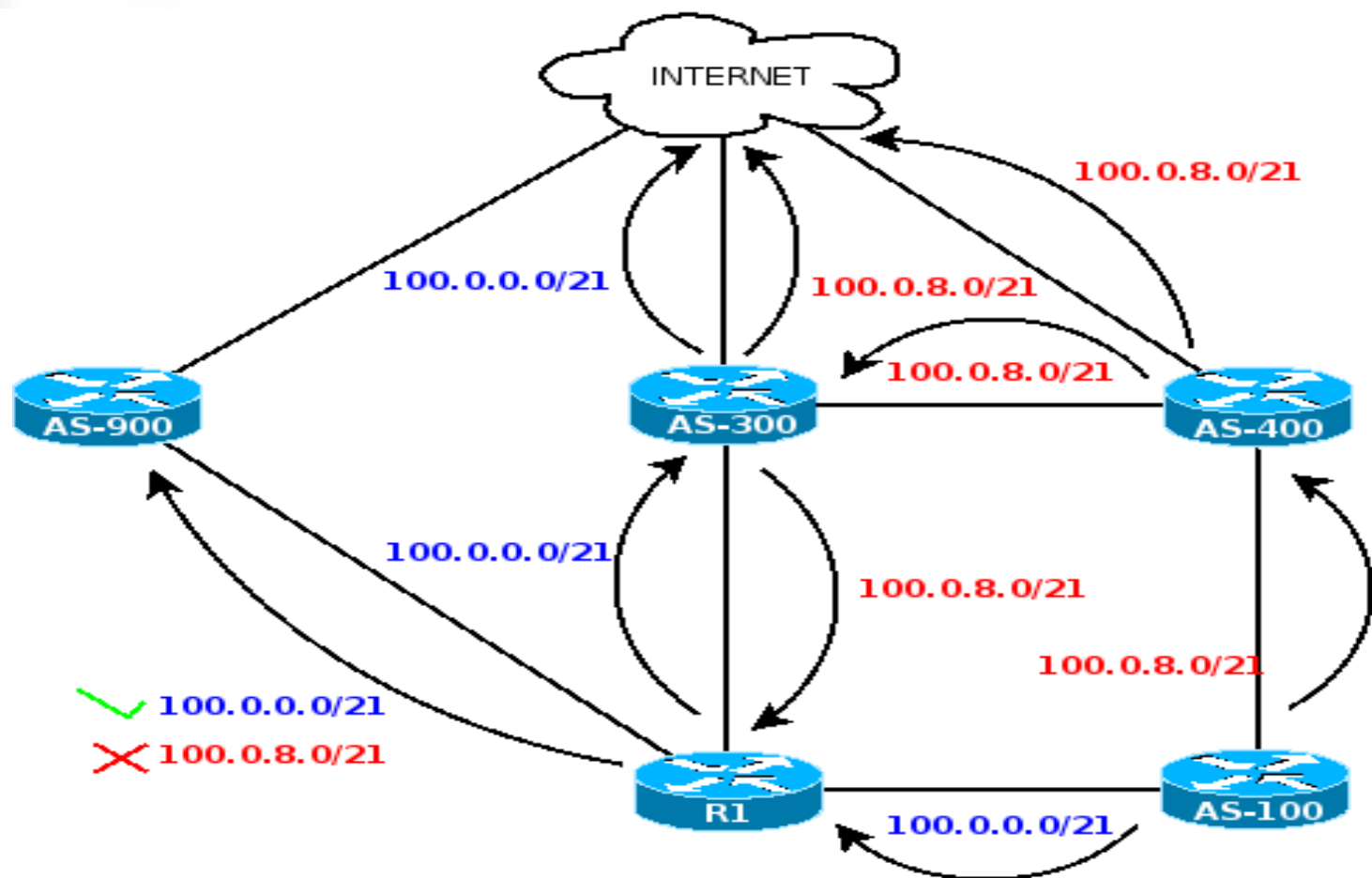
Um filtro similar ao do slide anterior "as100-in" no "**operadora-out**" vai funcionar mas deixa espaço para um problema:

Relembrando os primeiros slides sobre anúncios para mais de uma operadora, iremos visualizar a situação do ponto de vista da operadora, que agora somos nós e analisar a seguinte situação:

\*Meu cliente anuncia 100.0.0.0/21 para mim e 100.0.8.0/21 para sua outra operadora, e logo minha tabela haveria uma rota apontando para meu cliente (100.0.0.0/21) e outra para minha operadora (100.0.8.0/21), e meu router acabaria fazendo os dois anúncios para minha vizinhança como no slide a seguir...

### 3-Boas práticas para um AS de transito

#### 3.2 - Evitando propagar anúncios do seu cliente, mas que foram feitos para outra operadora.



## 3-Boas práticas para um AS de transito

### 3.2 - Evitando propagar anúncios do seu cliente, mas que foram feitos para outra operadora.

```
[admin@R1] /ip route print detail where dst-address in 100.0.0.0/20
```

```
0 ADb dst-address=100.0.0.0/21 gateway=192.168.0.2  
  gateway-status=192.168.0.2 reachable ether2-cliente distance=20  
scope=40 target-scope=10  
  bgp-as-path="100" bgp-med=0 bgp-origin=igp received-  
from=cliente_as100
```

```
1 ADb dst-address=100.0.8.0/21 gateway=200.200.200.1 gateway-  
status=200.200.200.1 reachable ether1-operadora_as300  
distance=20 scope=40 target-scope=10 bgp-as-path="300,400,100" bgp-  
origin=igp received-from=peer_operadora_as300
```

Observem no "bgp-as-path" que a rota #1 não foi recebida do meu cliente, mas sim da minha operadora A, e por estar sendo uma rota válida em minha tabela, vai ser propagado para todos os meus peers

## 3-Boas práticas para um AS de transito

### 3.2 - Evitando propagar anúncios do seu cliente, mas que foram feitos para outra operadora.

Nesses casos precisamos adicionar mais uma condição de match com o que chamamos de expressões regulares:

```
/routing filter>  
add action=accept chain=peer-operadora_as900-out prefix=100.0.0.0  
/20 prefix-length=20-24 bgp-as-path=^100$
```

\*traduzinho: quero receber a rede 100.0.0.0/20 ou qualquer sub-bloco /21, /22, /23 ou /24 mas apenas que comecem com "100" no seu as-path

Evitamos assim que anúncios recebidos por outra operadora sejam repassados indevidamente

Caso eu substitua "\$" por "\_" preservo ao meu cliente a liberdade de utilizar um "as-path-prepend" aceitando um AS-path assim: "...\_100, 100,100^"

\*O AS-Path é lido de trás para frente

## **3-Boas práticas para um AS de transito**

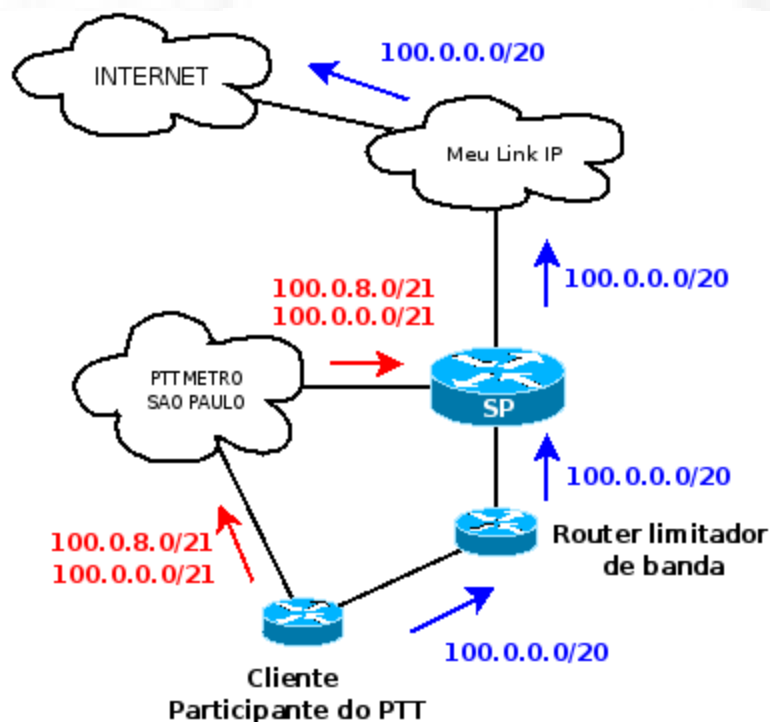
### **3.3 - AS's de trânsito nas trocas de tráfego**

Existem alguns pontos importantes à serem considerados, pois a ausência de alguns filtros podem causar prejuízos, aumento de latência ou loops de roteamento.

Um bom exemplo disso aconteceu com uma grande operadora que participa do PTT-SP.

## 3-Boas práticas para um AS de transito

### 3.3 - AS's de trânsito nas trocas de tráfego



O que aconteceria com o transito IP desse cliente?

## 3-Boas práticas para um AS de transito

### 3.3 - AS's de trânsito nas trocas de tráfego

Rotas recebidas diretamente pelo cliente

Rotas recebidas pela troca de tráfego

<i>100.0.0.0/20 =&gt; as-patch 100</i>	<i>1#</i>
<u><i>100.0.0.0/21 =&gt; as-patch 100</i></u>	<u><i>2#</i></u>
<u><i>100.0.8.0/21 =&gt; as-patch 100</i></u>	<u><i>3#</i></u>

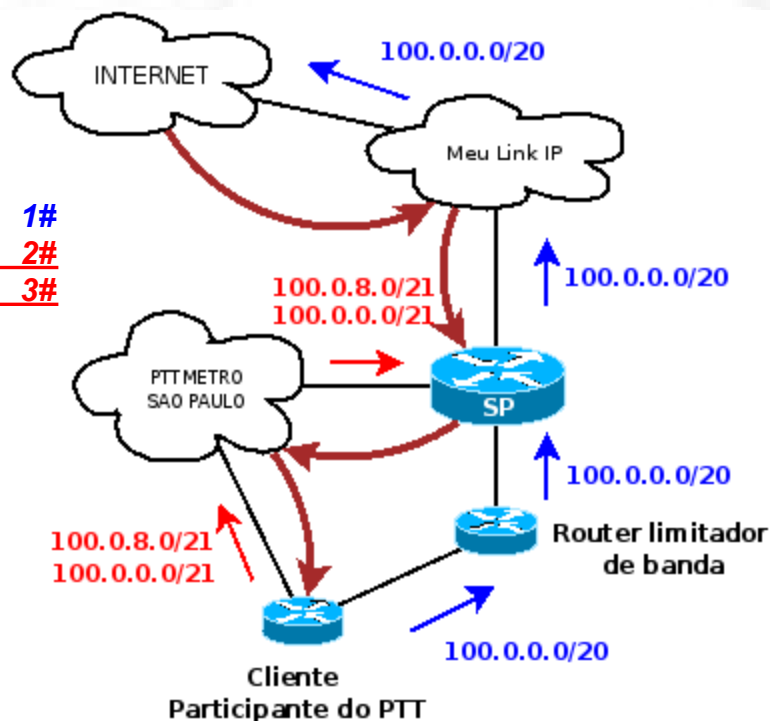
Sendo assim, todo transito IP com destino ao cliente, originado da internet, chegará ao seu router de borda, consumirá a banda da sua operadora, mas chegará até ele pela troca de tráfego, e com um certo conhecimento, o cliente pode manipular certos anúncios e fazer parecer que ele não está consumindo quase nenhum trânsito IP, quando na verdade, esse tráfego está "driblando" seus equipamentos que controlam banda.



## 3-Boas práticas para um AS de transito

### 3.3 - AS's de trânsito nas trocas de tráfego

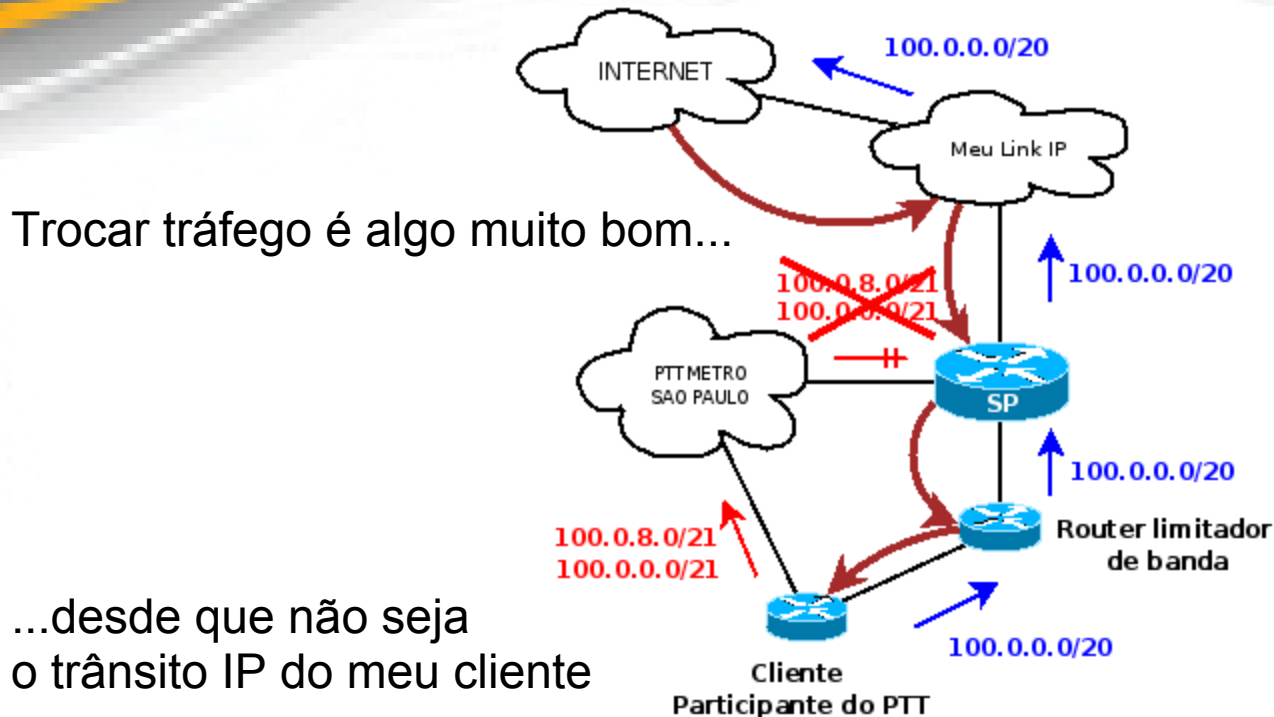
100.0.0.0/20 => as-patch 100 1#  
100.0.0.0/21 => as-patch 100 2#  
100.0.8.0/21 => as-patch 100 3#



Como evitar esse problema?

## 3-Boas práticas para um AS de transito

### 3.3 - AS's de trânsito nas trocas de tráfego



Solução:

Não receber nenhum anúncio que tenha sido originado pelo meu cliente no PTT

## 3-Boas práticas para um AS de transito

### 3.3 - AS's de trânsito nas trocas de tráfego

Para resolver isso, não devemos receber pela troca de tráfego anúncios de clientes que me comprem banda IP

```
/routing filter>
```

```
add action=discard chain=ptt-in bgp-as-patch=_100_
```

Qualquer rede anunciada pelo meu cliente na troca de tráfego será descartada pelo meu router

## 3-Boas práticas para um AS de transito

### 3.3 - Utilizando AS-Patch-Prepend

Como vimos na parte 1, é possível utilizar um determinado link apenas como "backup" se fizermos um anúncio menos específico para ele, de modo que esse link seria utilizado apenas caso o anúncio mais específico deixasse de ser propagado.

Porém como AS de trânsito não tenho como determinar qual bloco meu cliente me anunciará, e para fazer um link ser backup preciso aumentar o tamanho do AS-Patch para forçar os routers da internet a não escolher essa rota

```
/routing filter>
```

```
add action=accept chain=operadora-backup-out set-bgp-  
prepend=3
```

## 3-Boas práticas para um AS de transito

### 3.3 - Utilizando AS-Patch-Prepend

Rotas chegando na internet sem prepend

100.0.0.0/20 => as-patch 550,650,750,850,950, <b>1234</b> ,100	1#
<b>100.0.0.0/20 =&gt; as-patch 150,250,350,450,1234,100</b>	<b>2#</b>

Para alcançar 100.0.0.1 o roteador vai escolher a rota #2

\*\*\*\*\*  
\*\*\*

Rotas chegando na internet com prepend=3

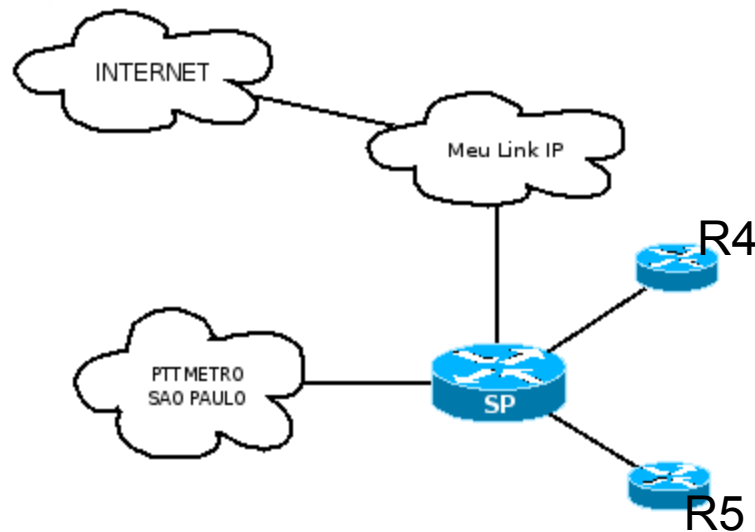
100.0.0.0/20 => as-patch 550,650,750,850,950, <b>1234</b> ,100	1#
100.0.0.0/20 => as-patch 150,250,350,450, <b>1234,1234,1234</b> ,100	2#

Agora para alcançar 100.0.0.1 o roteador vai escolher a rota #1

Dessa maneira interferimos na decisão de roteamento de um router que não temos gerência.

## 4-Utilizando BGP communities

### 4.1 - Identificando e exportando rotas recebidas de um determinado peer



Vamos imaginar agora um cenário mais complexo, onde temos vários POPs cada um com seu próprio link IP e vários clientes. Como distribuir apenas determinadas rotas entre esses POPs? (Por exemplo, as rotas recebidas do PTT-SP)

## 4-Utilizando BGP communities

### 4.1 - Identificando e exportando rotas recebidas de um determinado peer

Configurando:

```
/routing filter>  
add action=passthrough chain=ptt-in set-bgp-communities=1234:  
1234  
add action=accept chain=ibgp-out bgp-communities=1234:1234  
add action=discard chain=ibgp-out
```

Desse modo, apenas as rotas recebidas do peer PTT serão exportadas para todos os outros POPs do meu AS

```
[admin@R5] > ip route print detail where dst-address=8.8.8.0/24  
1 ADb dst-address=8.8.8.0/24 gateway=10.0.0.1  
    gateway-status=10.0.0.1 reachable ether1-link-sao_paulo  
distance=200 scope=40 target-scope=10 bgp-as-path="15169" bgp-  
origin=igp bgp-communities=1234:1234 received-from=peer_SPO
```

## 4-Utilizando BGP communities

### 4.4 - Manipulando a tabela de roteamento de outros roteadores na internet

Imagine ter que entrar em contato com o responsável de um determinado AS para solicitar uma alteração temporária e/ou de emergência. Uma community é uma código pre-definido por um AS que realizará automaticamente uma mudança em atributos das rotas que forem recebidas com uma determinada community agregada

Ex:

Eu como responsavel pelo AS 1234 aviso publicamente que para todas as rotas enviadas para mim com a comunidade 1234:90 agregada meus roteadores mudarão o LOCAL\_PREFERENCE dessa rota para 90

```
/routing filter>  
add action=passthrough chain=CLIENTE-IN bgp-communities=1234:  
90 set-bgp-local-pref=90
```

Assim, meus clientes podem manipular minha tabela setando para 90 o atributo local\_preference das rotas que eu recebo deles



# 4-Utilizando BGP communities

## 4.4 - Manipulando a tabela de roteamento de outros roteadores na internet

Exemplo de communities divulgadas pelas operadoras

### 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.

### Embratel

- 4230:120 - Local Preference - marcar a rota como localpreference 120
- 4230:10000 - Blackhole - bloqueia todo o trafego para a rede/endereco
- 4230:10002 - Blackhole - filtra o trafego 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

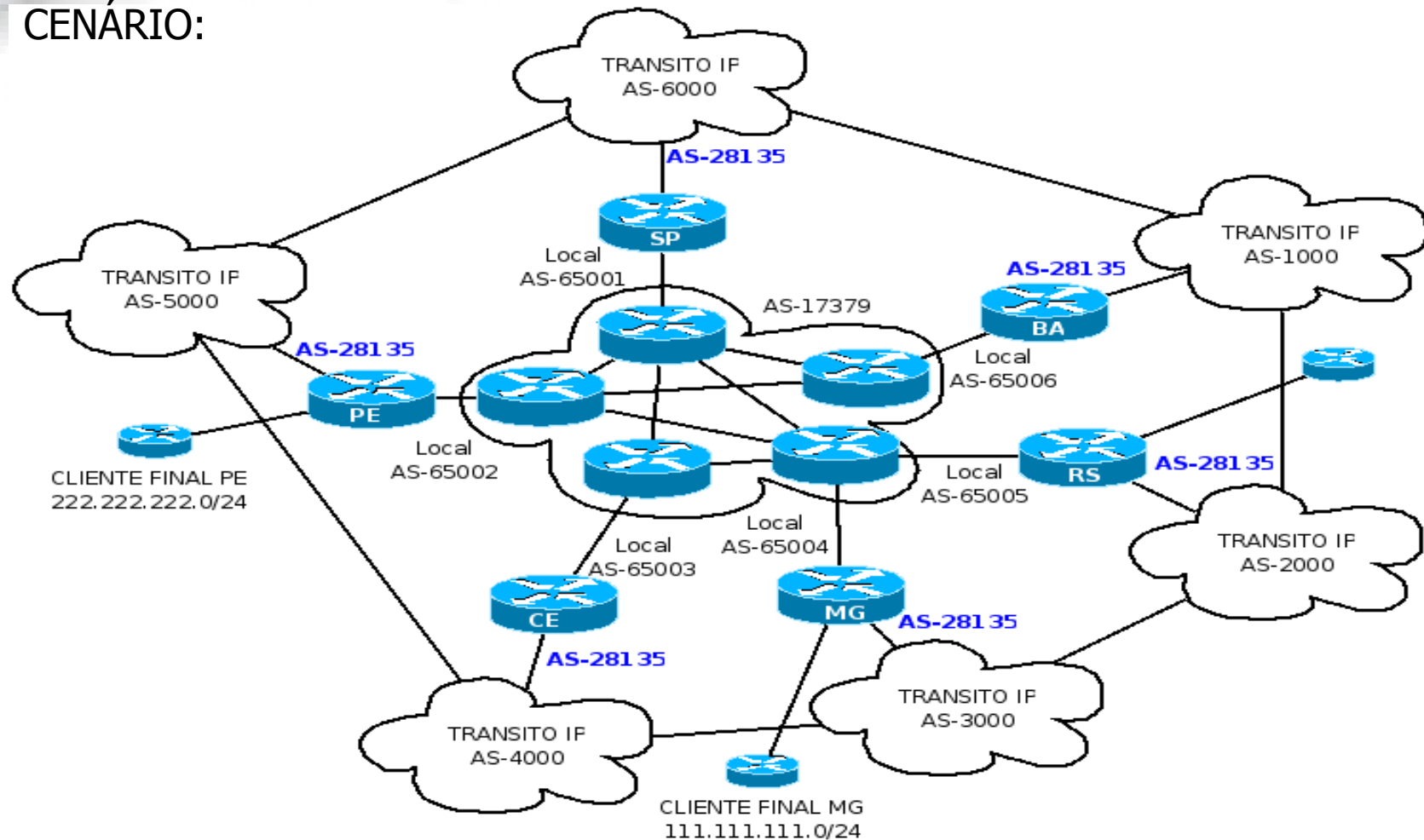
Para mais informações:

<http://eng.registro.br/pipermail/gter/2006-December/012791.html>

# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

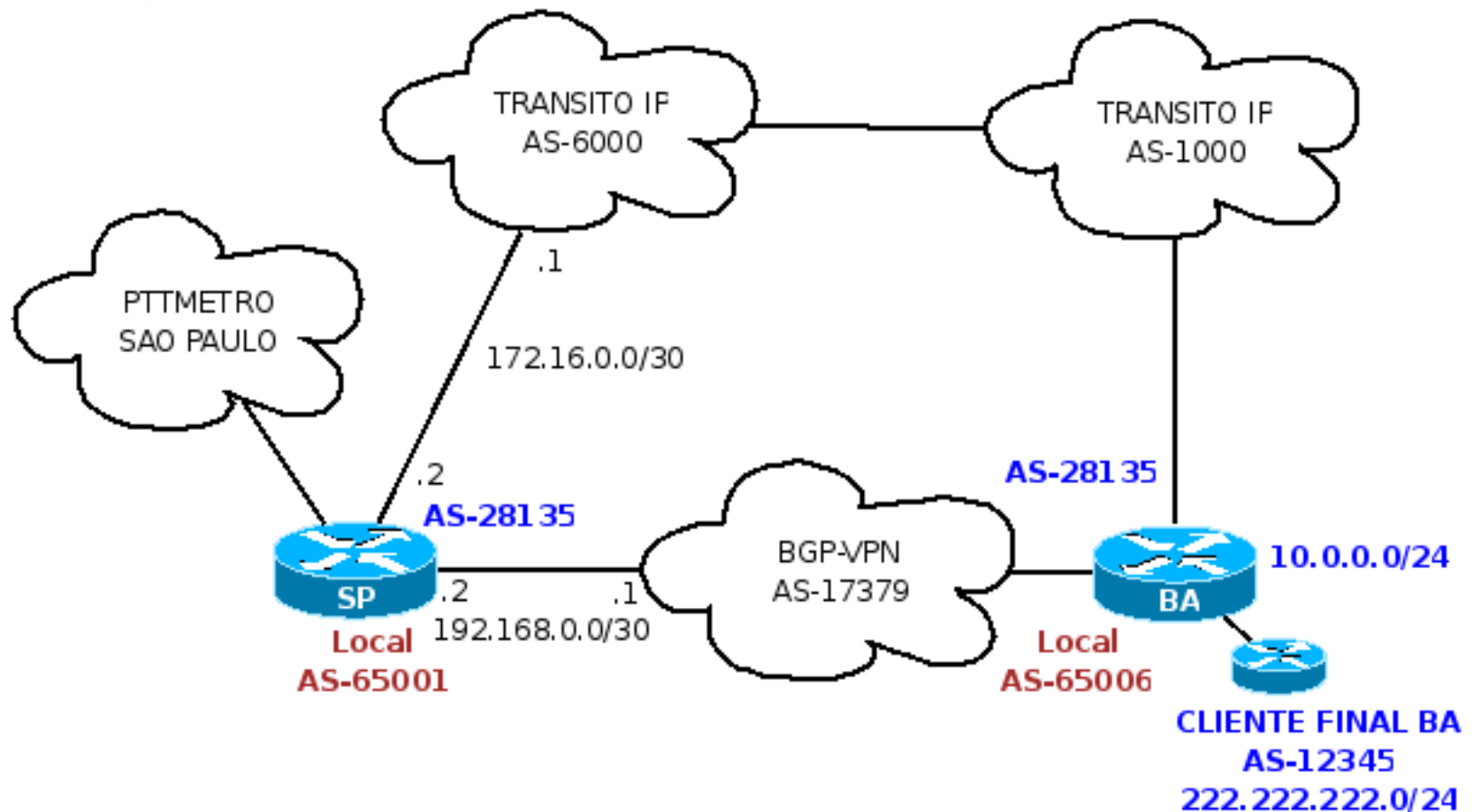
CENÁRIO:



## 5-Um estudo de caso

### 5.1 - Aplicação prática dos conceitos

Vamos focar nossa atenção em um POP para facilitar o entendimento:



# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

```
[admin@SP] > ip route print detail where dst-address =10.0.0.0/24
```

```
0 ADb dst-address=10.0.0.0/24 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether2-link_vpn distance=20  
scope=40 target-scope=10 bgp-as-path="17379,65006" bgp-origin=igp  
received-from=peer_vpn
```

```
1 Db dst-address=10.0.0.0/24 gateway=172.16.0.1  
gateway-status=172.16.0.1 reachable ether1-link_ip distance=20  
scope=40 target-scope=10 bgp-as-path="6000,1000,28135" bgp-origin=igp  
received-from=peer_transito_ip
```

```
[admin@SP] > ip route print detail where dst-address =222.222.222.0/24
```

```
0 ADb dst-address=222.222.222.0/24 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether2-link_vpn distance=20  
scope=40 target-scope=10 bgp-as-path="17379,65006,12345" bgp-origin=igp  
received-from=peer_vpn
```

```
1 Db dst-address=222.222.222.0/24 gateway=172.16.0.1  
gateway-status=172.16.0.1 reachable ether1-link_ip distance=20  
scope=40 target-scope=10 bgp-as-path="6000,1000,28135,12345" bgp-origin=igp  
received-from=peer_transito_ip
```

# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

[admin@CLIENTE\_ANID > ip route print detail where dst-address =10.0.0.0/24

0 Db dst-address=10.0.0.0/24 gateway=172.16.200.1  
gateway-status=172.16.200.1 reachable ether1-link\_anid distance=20  
scope=40 target-scope=10 bgp-as-path="28135,17379,65006" bgp-origin=igp  
received-from=peer\_vpn

1 ADb dst-address=10.0.0.0/24 gateway=200.200.200.1  
gateway-status=200.200.200.1 reachable ether1-link\_as1000 distance=20  
scope=40 target-scope=10 bgp-as-path="1000,28135" bgp-origin=igp  
received-from=peer\_as1000

[admin@CLIENTE\_ANID > ip route print detail where dst-address =222.222.222.0/24

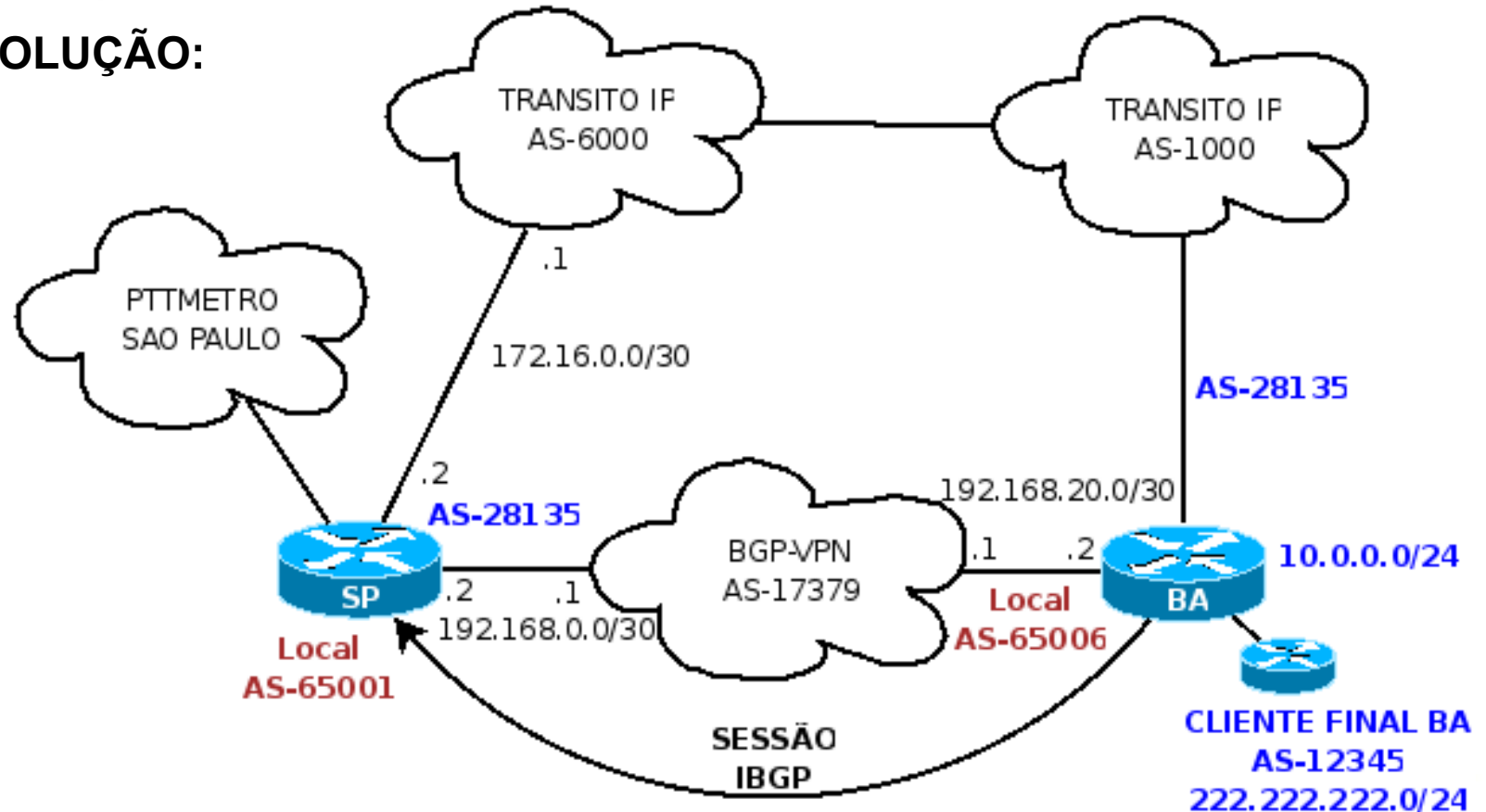
0 Db dst-address=222.222.222.0/24 gateway=172.16.200.1  
gateway-status=172.16.200.1 reachable ether1-link\_anid distance=20  
scope=40 target-scope=10 bgp-as-path="28135,17379,65006,12345" bgp-origin=igp  
received-from=peer\_vpn

1 ADb dst-address=222.222.222.0/24 gateway=200.200.200..1  
gateway-status=200.200.200.1 reachable ether1-link\_as1000 distance=20  
scope=40 target-scope=10 bgp-as-path="1000,28135,12345" bgp-origin=igp  
received-from=peer\_as1000

## 5-Um estudo de caso

### 5.1 - Aplicação prática dos conceitos

SOLUÇÃO:



## 5-Um estudo de caso

### 5.1 - Aplicação prática dos conceitos

Configurando:

```
admin@SP] /routing bgp peer>
```

```
add name=ibgp-bahia remote-address=192.168.20.2 remote-as=28135  
in-filter=ibgp-in out-filter=ibgp-out nexthop-choice=force-self
```

```
admin@BA] /routing bgp peer>
```

```
add name=ibgp-sao_paulo remote-address=192.168.0.2 remote-  
as=28135 in-filter=ibgp-spo-in out-filter=ibgp-spo-out nexthop-  
choice=force-self
```

# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

Exemplo das rotas no router de São Paulo após sessão IBGP

```
[admin@SP] > ip route print detail where dst-address =10.0.0.0/24
```

```
0 ADb dst-address=10.0.0.0/24 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether2-link_vpn distance=20  
scope=40 target-scope=10 bgp-as-path="17379,65006" bgp-origin=igp  
received-from=peer_vpn
```

```
1 Db dst-address=10.0.0.0/24 gateway=172.16.0.1  
gateway-status=172.16.0.1 reachable ether1-link_ip distance=20  
scope=40 target-scope=10 bgp-as-path="6000,1000,28135" bgp-origin=igp  
received-from=peer_transito_ip
```

```
2 Db dst-address=10.0.0.0/24 gateway=192.168.20.2  
gateway-status192.168.20.2 unreachable distance=200 scope=40  
target-scope=30 bgp-local-pref=100 bgp-med=0 bgp-origin=igp  
received-from=ibgp_bahia
```

PS: Caso haja uma rota padrão instalada, o RouterOS troca o "**unreachable**" por "**recursive via x.x.x.x**" onde x.x.x.x é o nexthop da rota padrão, o que na prática dá no mesmo já que o router mandaria os pacotes para a sua rota padrão, o que não nos interessa.



# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

Exemplo das rotas no router de São Paulo após sessão IBGP

```
[admin@SP] > ip route print detail where dst-address =222.222.222.0/24
```

- 0 ADb dst-address=222.222.222.0/24 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether2-link\_vpn distance=20  
scope=40 target-scope=10 bgp-as-path="17379,65006,12345" bgp-origin=igp  
received-from=peer\_vpn
- 1 Db dst-address=222.222.222.0/24 gateway=172.16.0.1  
gateway-status=172.16.0.1 reachable ether1-link\_ip distance=20  
scope=40 target-scope=10 bgp-as-path="6000,1000,28135,12345" bgp-origin=igp  
received-from=peer\_transito\_ip
- 2 Db dst-address=222.222.222.0/24 gateway=192.168.20.2  
gateway-status=192.168.20.2 **unreachable** distance=200 scope=40  
**target-scope=30** bgp-as-path="12345" bgp-local-pref=100  
bgp-origin=igp received-from=ibgp-bahia

# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

### Resolvendo o problema da rota inalcançável:

Nesse caso, o nosso "next-hop" (192.168.20.2) não faz parte de uma rede diretamente conectada, mas sim de uma rede que foi recebida via BGP conforme print abaixo:

```
[admin@SP] > ip route print detail where dst-address =192.168.20.0/30
```

```
0 ADb dst-address=192.168.20.0/30 gateway=192.168.0.1  
gateway-status=192.168.0.1 reachable ether2-link_vpn distance=20 scope=40  
target-scope=10 bgp-as-path="17379" bgp-origin=igp received-from=peer_vpn
```

```
admin@SP] /routing filter>
```

```
add action=passthrough chain=ibgp-in set-target-scope=40 set-bgp-  
local-pref=120
```

# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

Exemplo das rotas no router de São Paulo após sessão IBGP estabelecida

```
[admin@SP] > ip route print detail where dst-address =10.0.0.0/24
```

- 0 Db dst-address=10.0.0.0/24 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether2-link\_vpn distance=20  
scope=40 target-scope=10 bgp-as-path="17379,65006" bgp-origin=igp  
received-from=peer\_vpn
- 1 Db dst-address=10.0.0.0/24 gateway=172.16.0.1  
gateway-status=172.16.0.1 reachable ether1-link\_ip distance=20  
scope=40 target-scope=10 bgp-as-path="6000,1000,28135" bgp-origin=igp  
received-from=peer\_transito\_ip
- 2 ADb dst-address=10.0.0.0/24 gateway=192.168.20.2  
gateway-status192.168.20.2 recursive via 192.168.0.1 distance=200 scope=40  
target-scope=40 bgp-local-pref=120 bgp-med=0 bgp-origin=igp  
received-from=ibgp\_bahia

PS: Como apenas a rota válida é repassada para os outros peers, o AS-Patch será dessa vez propagado corretamente sem que apareçam o AS privado nem o AS da minha operadora.

# 5-Um estudo de caso

## 5.1 - Aplicação prática dos conceitos

Exemplo das rotas no router de São Paulo após sessão IBGP

```
[admin@SP] > ip route print detail where dst-address =222.222.222.0/24
```

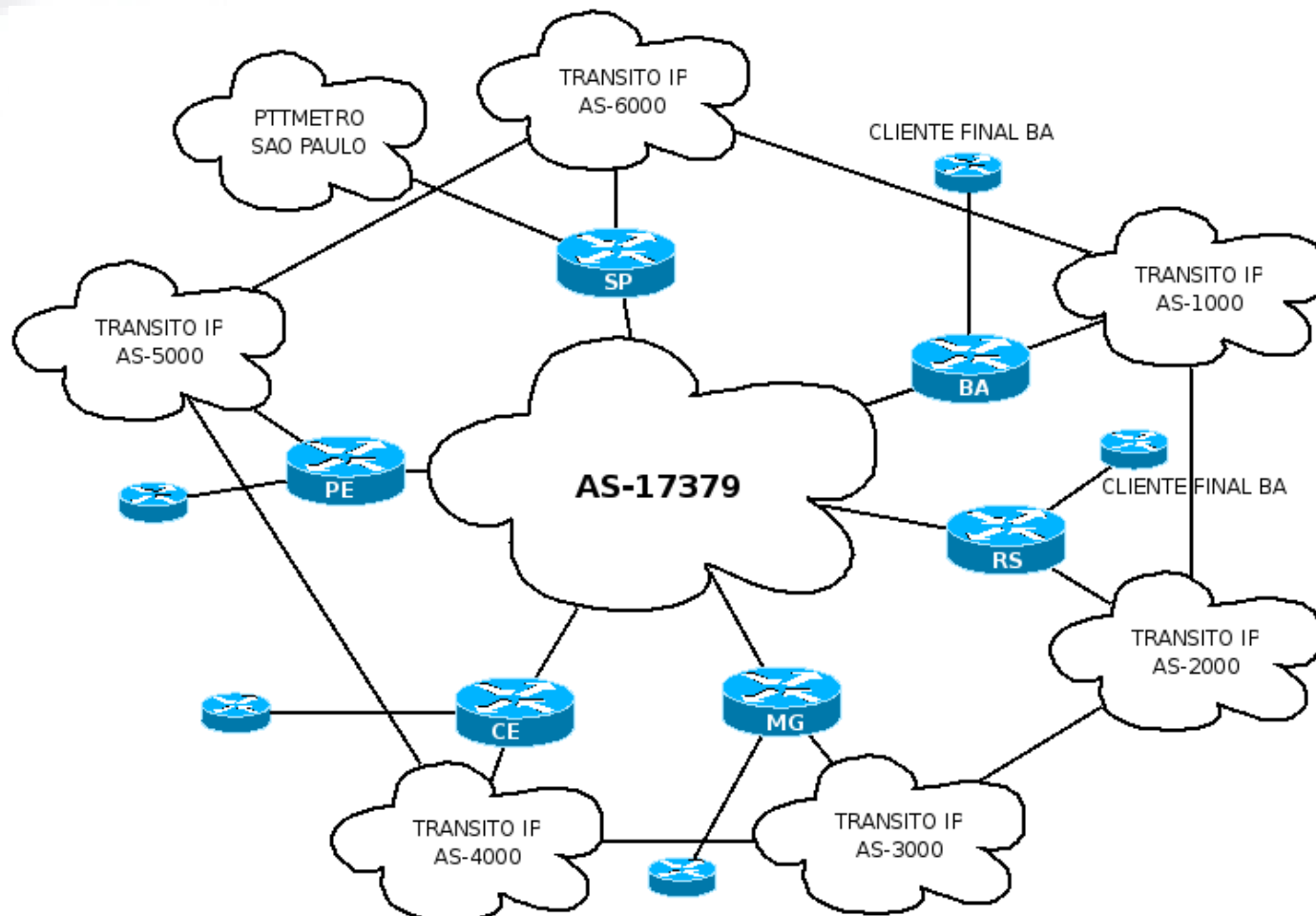
- 0 Db dst-address=222.222.222.0/24 gateway=192.168.1.1  
gateway-status=192.168.1.1 reachable ether2-link\_vpn distance=20  
scope=40 target-scope=10 bgp-as-path="17379,65006,12345" bgp-origin=igp  
received-from=peer\_vpn
- 1 Db dst-address=222.222.222.0/24 gateway=172.16.0.1  
gateway-status=172.16.0.1 reachable ether1-link\_ip distance=20  
scope=40 target-scope=10 bgp-as-path="6000,1000,28135,12345" bgp-origin=igp  
received-from=peer\_transito\_ip
- 2 ADb dst-address=222.222.222.0/24 gateway=192.168.20.2  
gateway-status=192.168.20.2 recursive via 192.168.0.1 distance=200 scope=40  
target-scope=40 bgp-as-path="12345" bgp-local-pref=300  
bgp-origin=igp received-from=ibgp-bahia

Na tabela dos clientes de transito IP ou participantes do PTT-Metro a lógica seria a mesma, e os AS-Patch indesejado não iria mais ser propagado.

# 5-Um estudo de caso

## 5.2 - Um problema de escalabilidade

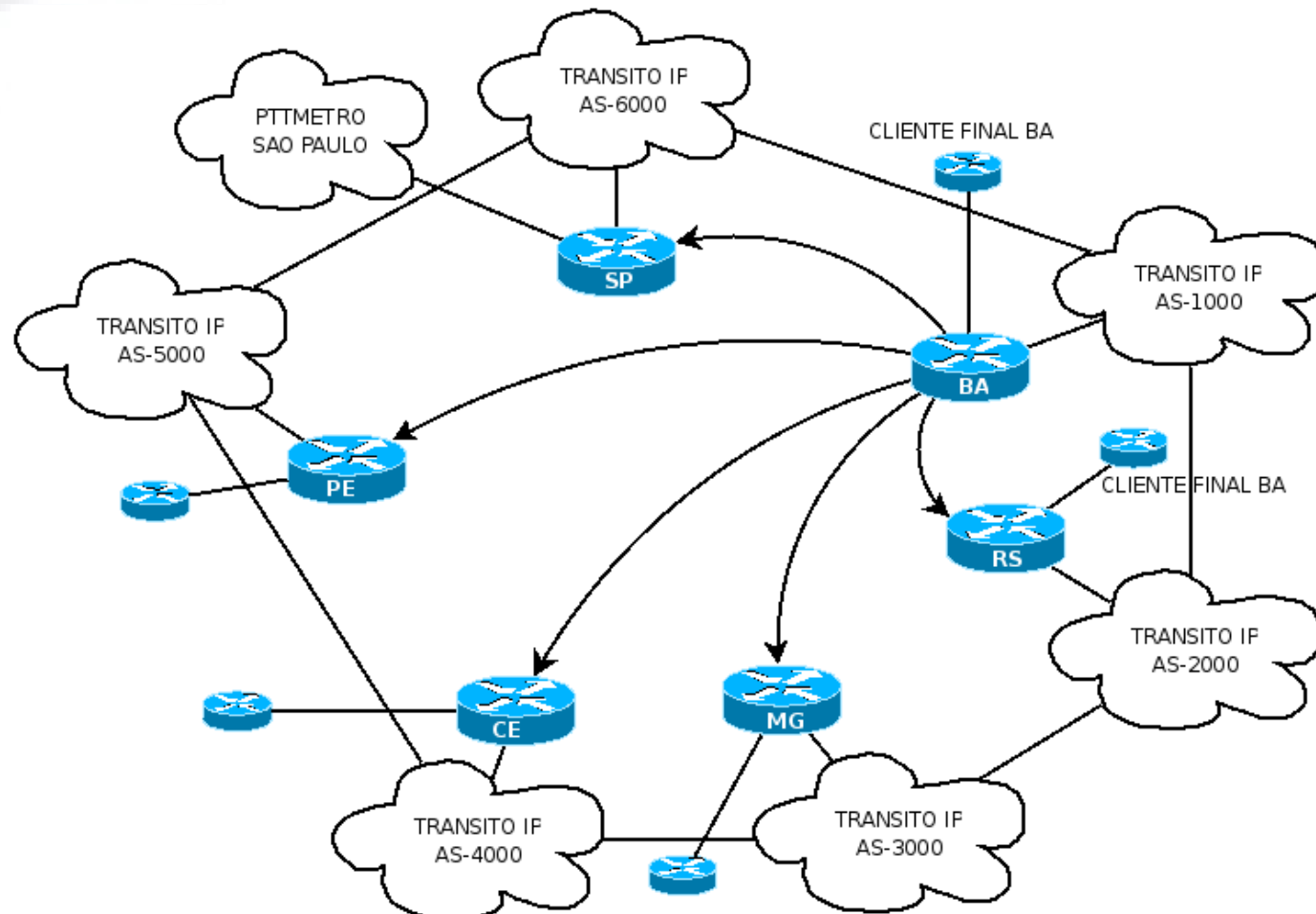
Estabelecendo sessões ibgp entre todos os POPs



# 5-Um estudo de caso

## 5.2 - Um problema de escalabilidade

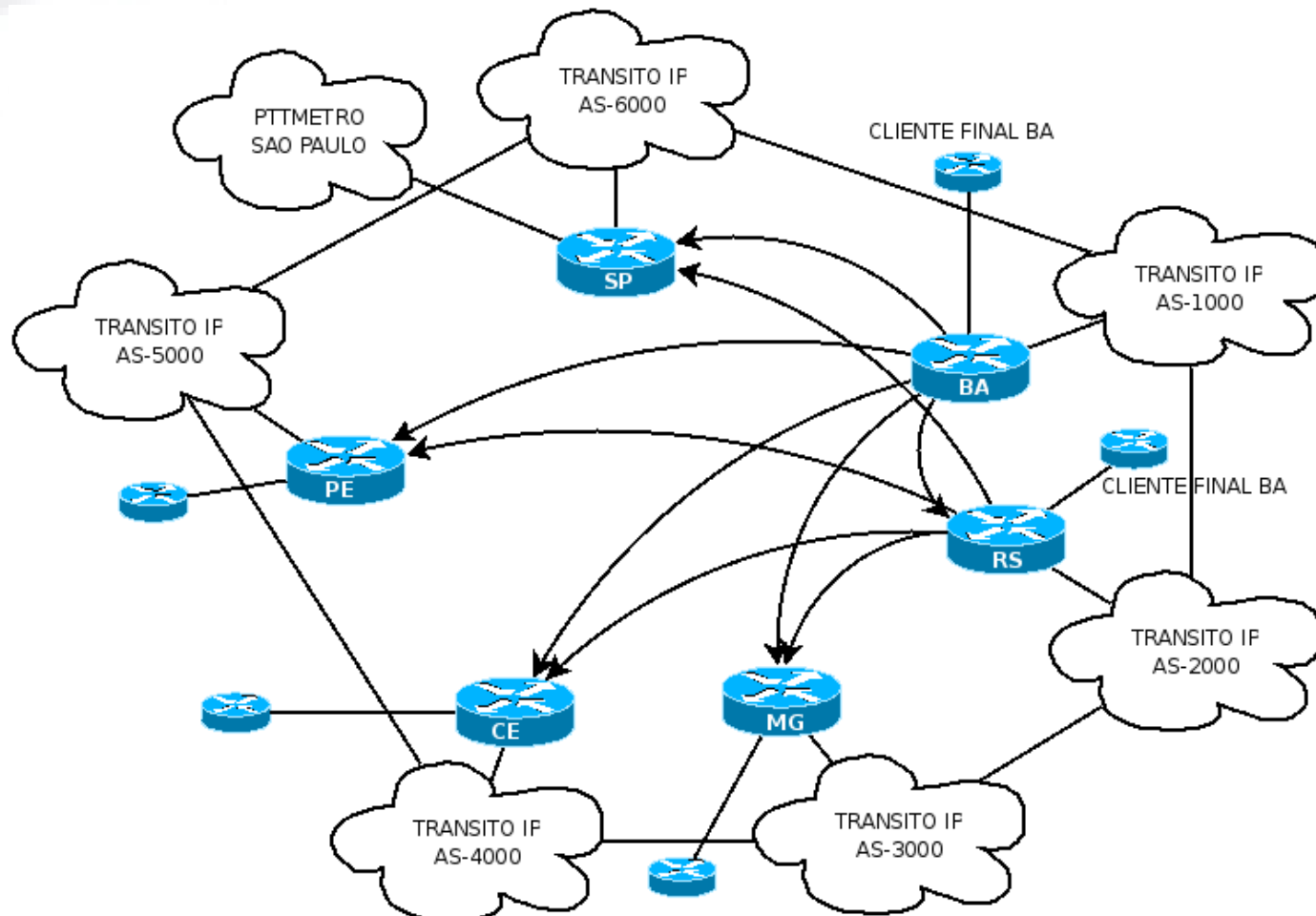
Começando por Bahia:



# 5-Um estudo de caso

## 5.2 - Um problema de escalabilidade

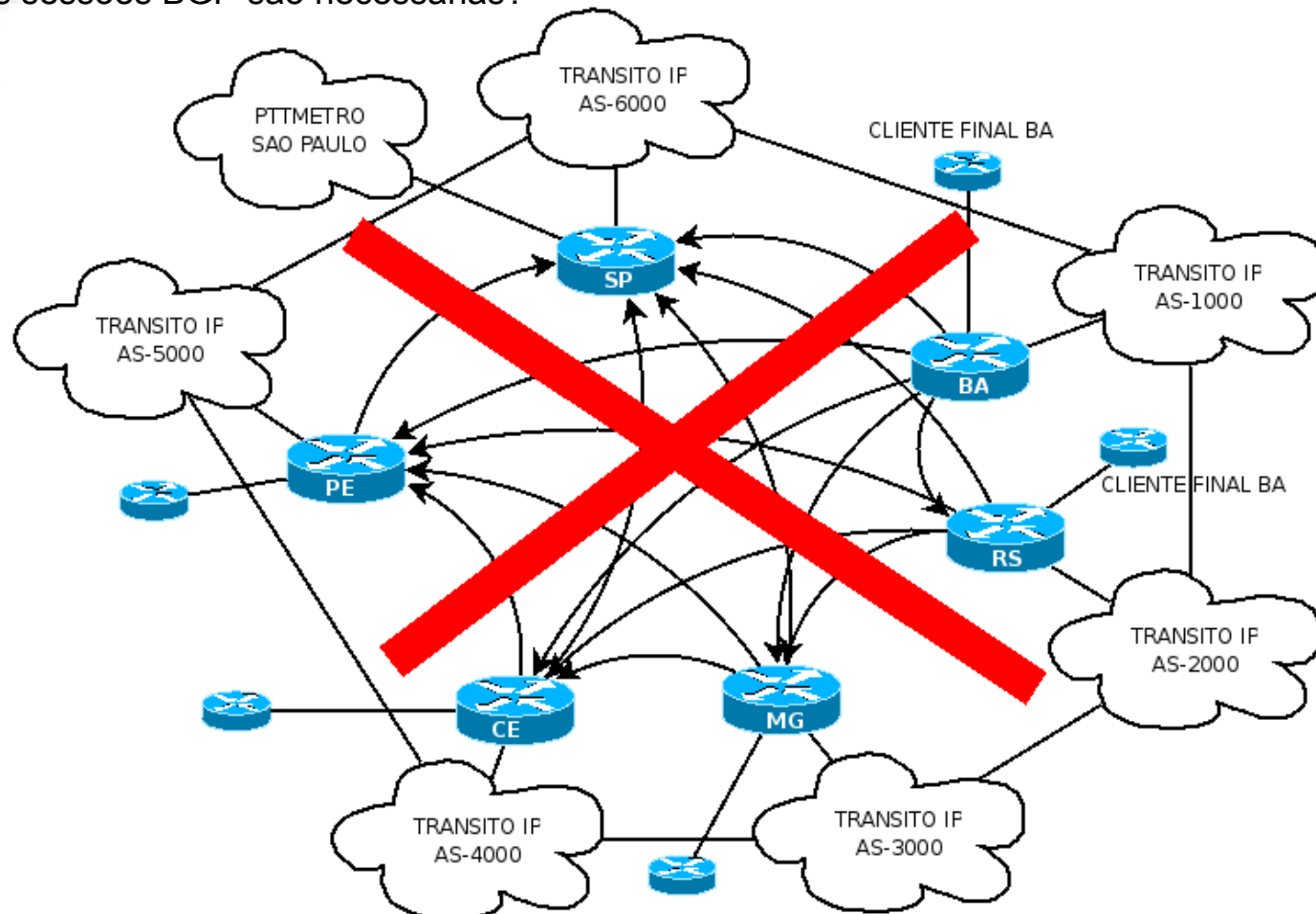
Depois Rio Grande do Sul:



# 5-Um estudo de caso

## 5.2 - Um problema de escalabilidade

Ficamos assim diante de uma dificuldade extrema, por exemplo, tenho 35 POPs ativos hoje, quantas sessões BGP são necessárias?





# 5-Um estudo de caso

## 5.2 - Um problema de escalabilidade

Para evitar algumas dezenas de slides vamos fazer uma análise combinatória enquanto nos resta tempo para "desistir"

Quantas sessões BGP no total?

vou chamar esse valor de **T**

6 POPs (elementos) que vou chamar de "**e**"

Sessão BGP = agrupamento 1+1 (sem repetições nem AA nem BA) que vou chamar de "**a**"

$$T(e,a) = e! / [ (e-a)! * a! ]$$

Aplicando a fórmula para os 6 POPs do exemplo

$$\text{fat}(6) / [ \text{fat}(6-2) * \text{fat}(2) ] = 15$$

Aplicando a fórmula na quantidade de POPs ativos na ANID

$$\text{fat}(35) / [ \text{fat}(35-2) * \text{fat}(2) ] = 595$$

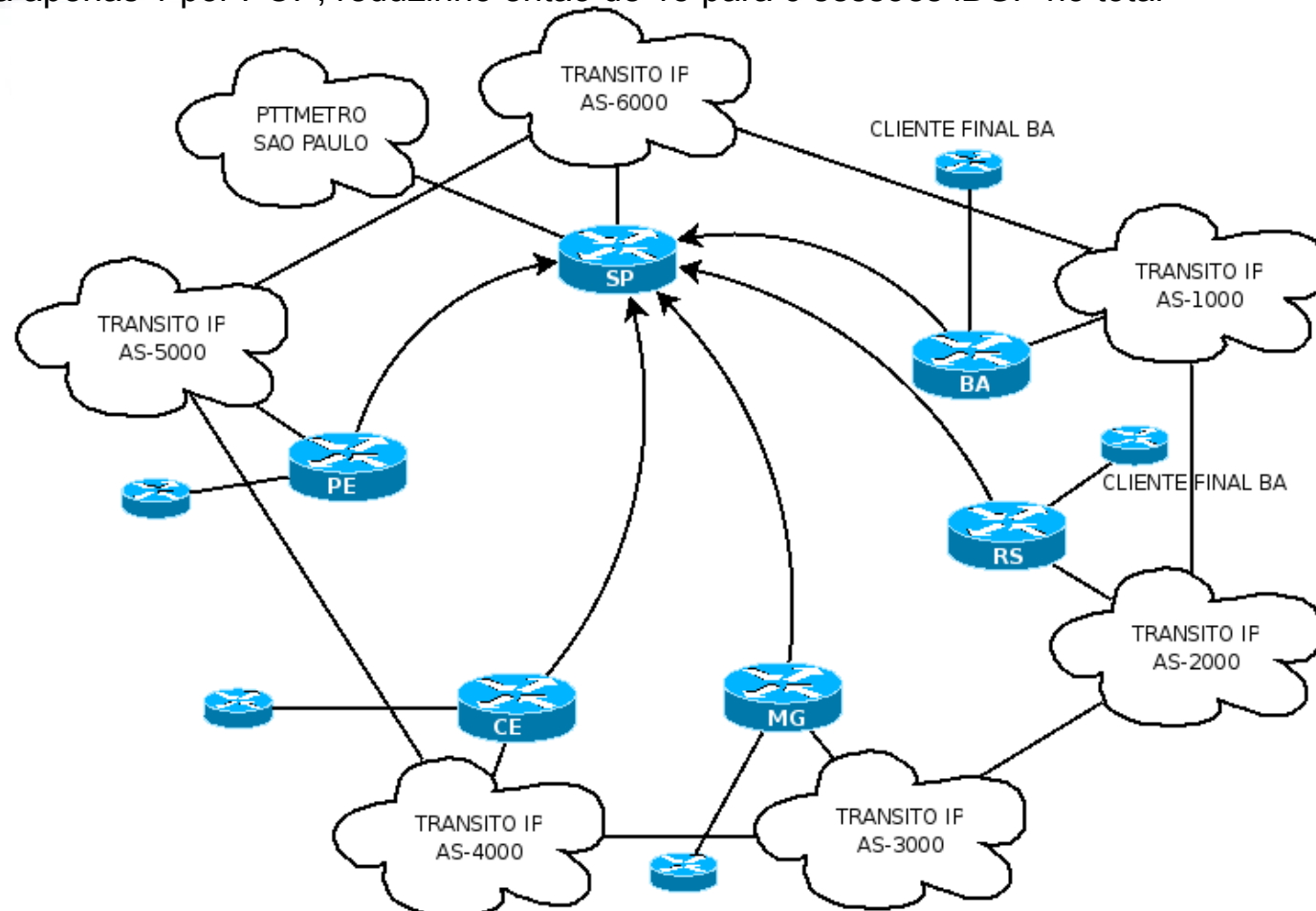
Aplicando ao meu planejamento futuro de 60 POPs

$$\text{fat}(60) / [ \text{fat}(60-2) * \text{fat}(2) ] = 1770$$

# 5-Um estudo de caso

## 5.2 - Um problema de escalabilidade

Com "Route-Reflector" em SP podemos reduzir a quantidade de sessões iBGP dentro do meu AS para apenas 1 por POP, reduzindo então de 15 para 6 sessões iBGP no total

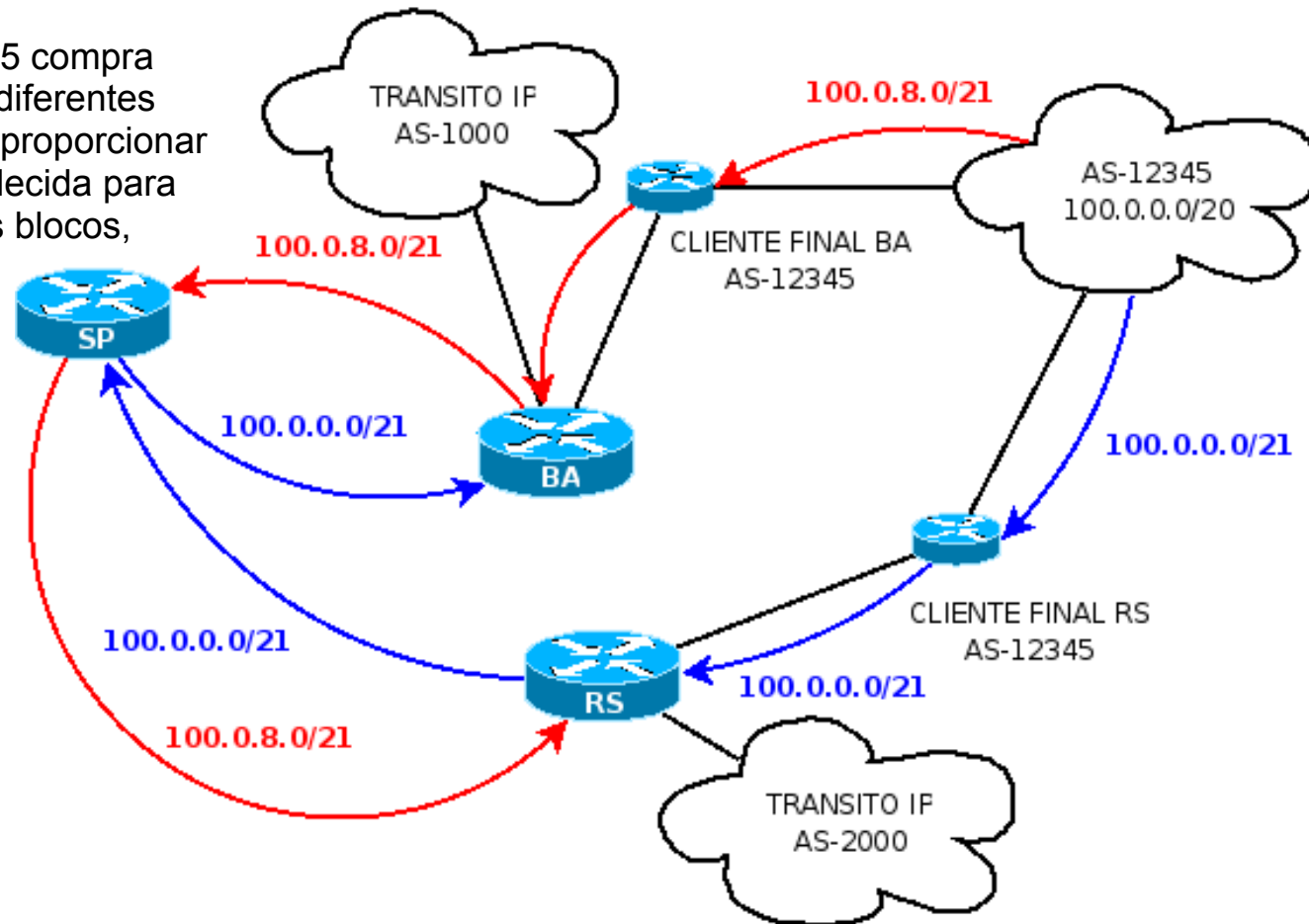


## 5-Um estudo de caso

### 5.3 - Uma solução e um novo problema

A solução de route-reflector resolve o meu problema de escalabilidade, porém me traz um novo problema em um cenário particular, porém relativamente comum:

Meu cliente do AS-12345 compra  
Transito IP em 2 POPs diferentes  
E meus filtros precisam proporcionar  
liberdade para que ele decida para  
qual POP anunciar seus blocos,  
bem como modificar  
esses anuncios quando  
bem entender



## 5-Um estudo de caso

### 5.3 - Uma solução e um novo problema

```
[admin@RS] /ip route print detail where dst-address in 100.0.0.0/20
```

```
0 ADb dst-address=100.0.0.0/21 gateway=172.16.40.2  
  gateway-status=172.168.40.2 reachable ether3-cliente distance=20  
scope=40 target-scope=10  
  bgp-as-path="12345" bgp-med=0 bgp-origin=igp received-  
from=cliente_as12345
```

```
1 ADb dst-address=100.0.8.0/21 gateway=192.168.20.2 gateway-  
status=192.168.20.2 recursive via 192.168.30.1 ether2-link_vpn  
distance=20 scope=40 target-scope=40 bgp-as-path="12345" bgp-  
origin=igp received-from=ibgp-sao_paulo
```

**Observem que nesse caso, o "bgp-as-path" permanece o mesmo, tanto na rota recebida diretamente do cliente (#0) quando da rota que o cliente anunciou no POP Bahia e foi "refletida" por SP, a condição de match utilizada na parte 3.2 irá anunciar erroneamente as duas rotas para o circuito IP local...**

## 5-Um estudo de caso

### 5.3 - Uma solução e um novo problema

Vamos analisar as condições de match:

```
/routing filter>
```

```
add action=accept chain=peerX-out prefix=100.0.0.0/20 prefix-  
length=20-24 bgp-as-path=^12345_
```

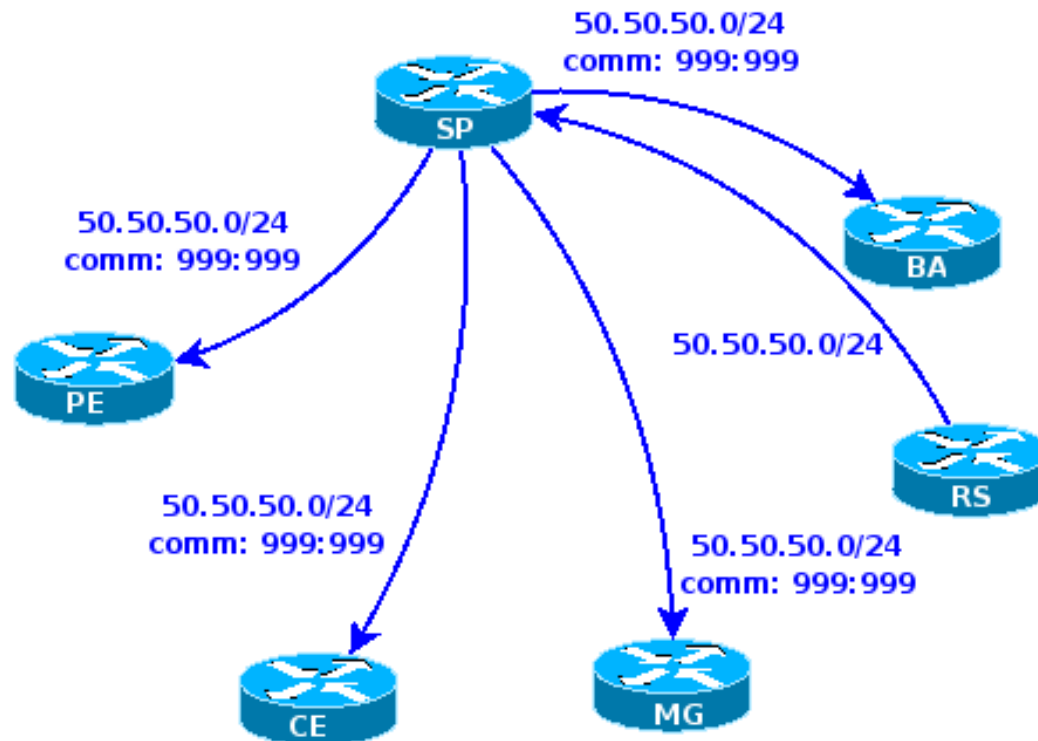
Note que as duas rotas começam com ^12345, já que a rota que não quero anunciar foi recebida de um router do meu próprio AS, e por ser do meu AS não houve incremento no AS-Patch.

O que fazer?

## 5-Um estudo de caso

### 5.3 - Uma solução e um novo problema

Existem várias maneiras, porém muito difíceis de escalar em sua maioria, vamos utilizar uma que exija o mínimo de configurações:



Faremos o Route Reflector inserir uma comunidade "999:999" em todas as rotas refletidas por iBGP

## 5-Um estudo de caso

### 5.3 - Uma solução e um novo problema

Configurando o Route-Reflector:

```
[admin@SP /routing filter>
```

```
add action=passthrough chain=ibgp-in set-bgp-communities=999:999
```

## 5-Um estudo de caso

### 5.3 - Uma solução e um novo problema

Vamos verificar agora como ficou a tabela

```
[admin@RS] /ip route print detail where dst-address in 100.0.0.0/20
```

```
0 ADb dst-address=100.0.0.0/21 gateway=172.16.40.2  
    gateway-status=172.168.40.2 reachable ether3-cliente distance=20  
scope=40 target-scope=10 bgp-as-path="12345" bgp-med=0 bgp-  
origin=igp received-from=cliente_as12345
```

```
1 ADb dst-address=100.0.8.0/21 gateway=192.168.20.2 gateway-  
status=192.168.20.2 recursive via 192.168.30.1 ether2-link_vpn  
distance=20 scope=40 target-scope=40 bgp-as-path="12345" bgp-  
origin=igp bgp-communities=999:999 received-from=ibgp-sao_paulo
```

Agora tenho como proibir o anúncio das rotas refletidas utilizando communities! Vejamos como:



## 5-Um estudo de caso

### 5.3 - Uma solução e um novo problema

Adicionando mais uma condição de match:

```
[admin@RS] /routing filter>
```

```
add action=accept chain=peerX-out prefix=100.0.0.0/20 prefix-length=20-24 bgp-as-path=^12345_ bgp-communities=!999:999
```

Ou inserindo no topo uma regra de discard:

```
[admin@RS] /routing filter>
```

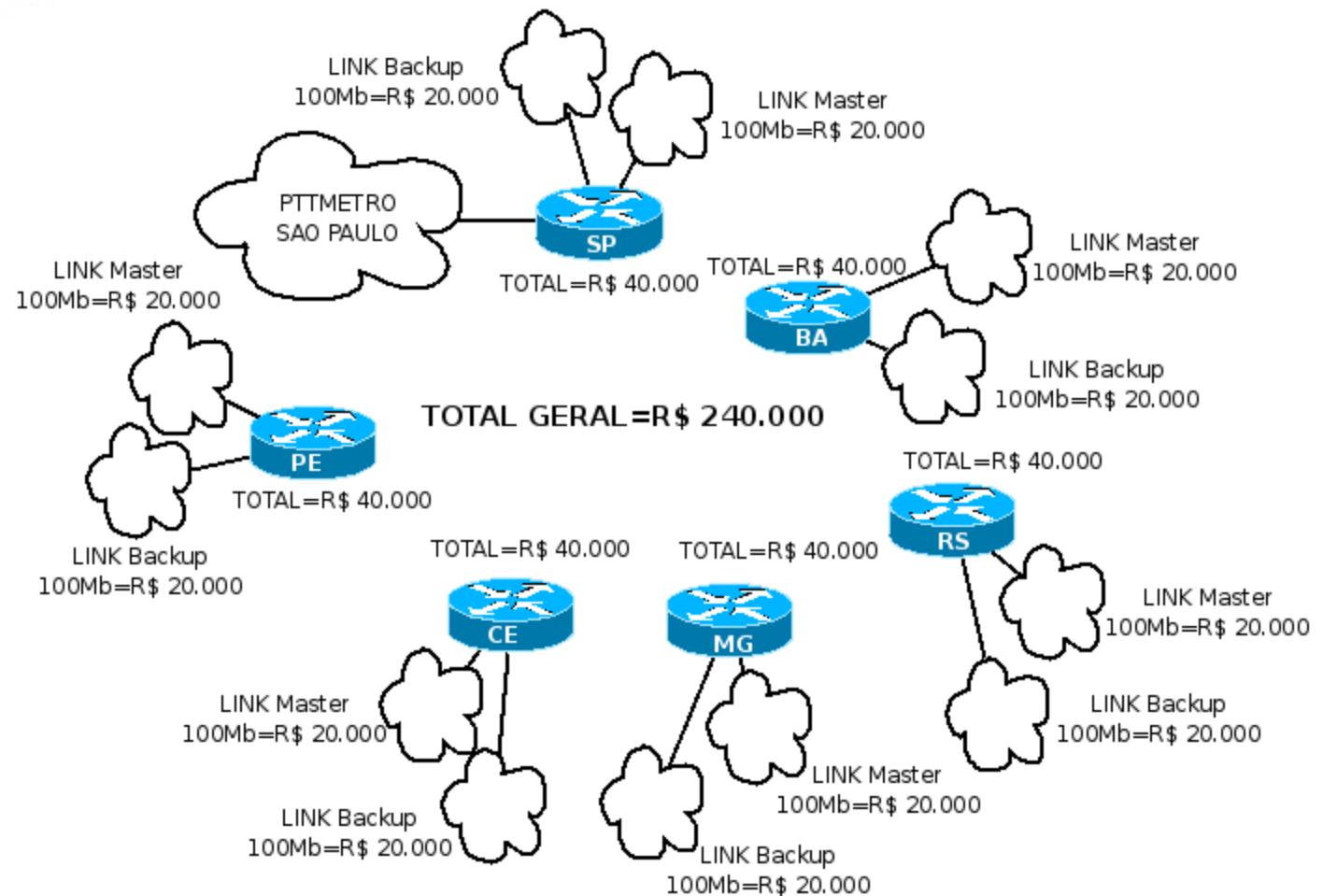
```
add action=discard chain=peerX-out bgp-communities=999:999
```

É interessante negar esse anúncio apenas para minha operadora local, e permitir para os meus clientes de transito IP

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Considere um link redundante para cada POP:



## **5-Um estudo de caso**

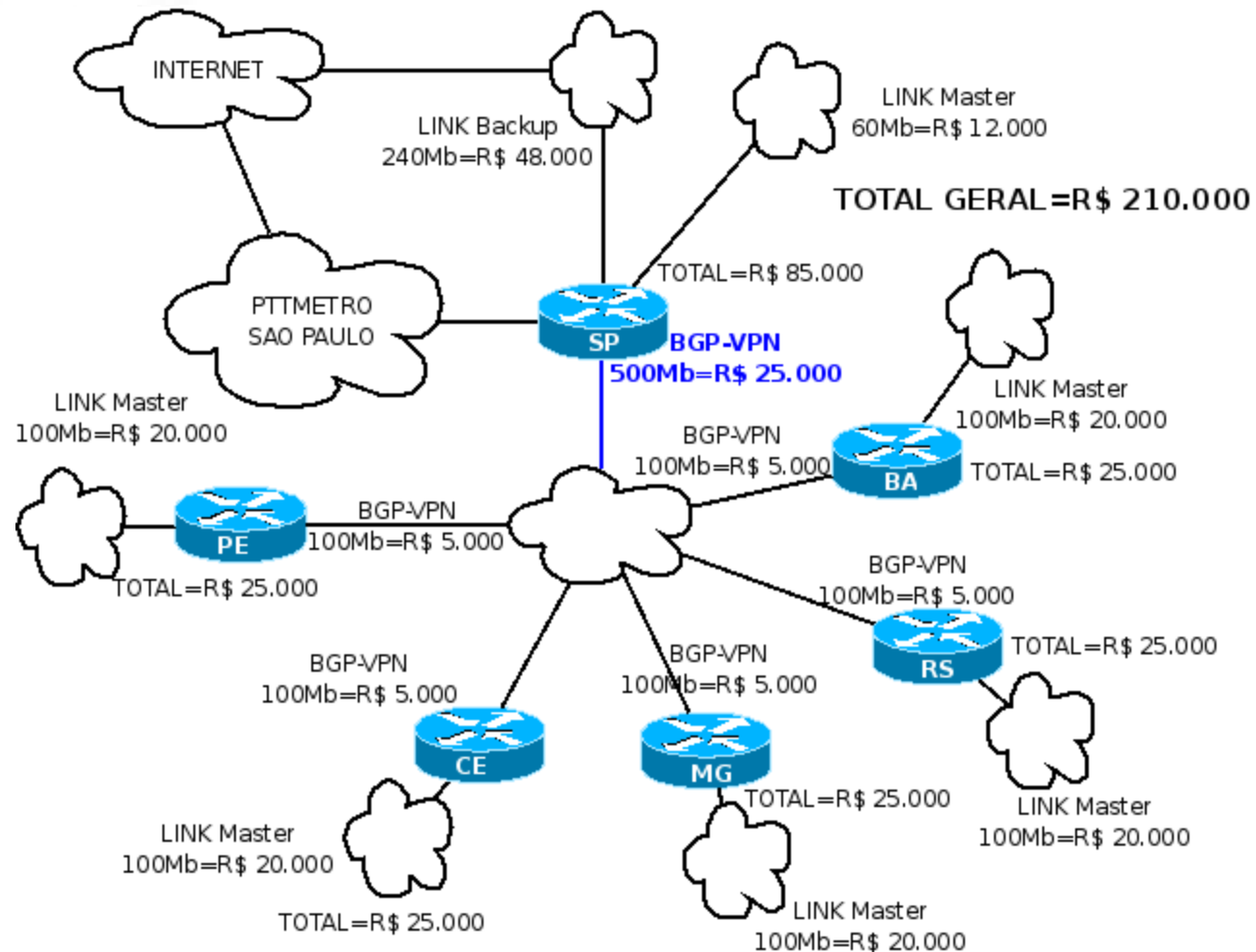
### **5.4 - Um modelo relativamente barato de link redundante**

No modelo do slide anterior há um grande desperdício de dinheiro levando em conta que dificilmente todos os links cairão simultaneamente, sendo assim, podemos diminuir os custos se conectando a um PIX do PTT-Metro-SP, comprando um link maior para backup em São Paulo e trazendo esse link através de circuitos de transporte BGP-VPN conforme veremos a seguir:

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Por que apenas 240Mb de link backup em SP e 500Mb VPN?



## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

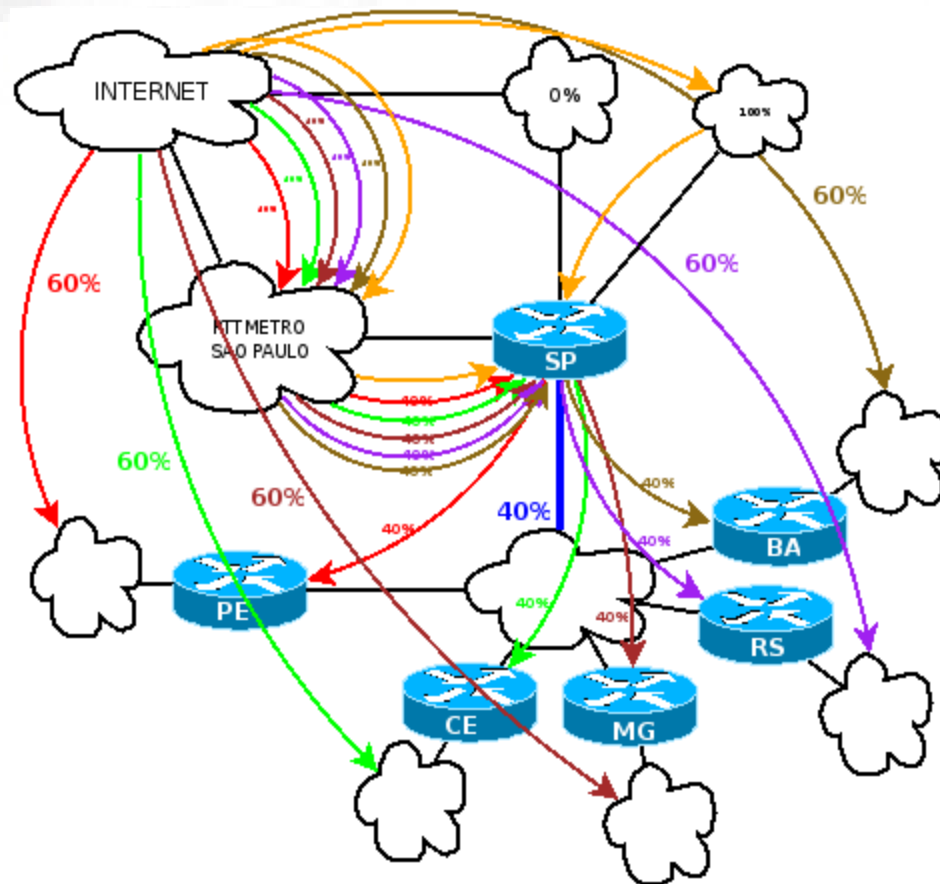
Segundo estatísticas, aproximadamente 40% do volume total de tráfego dos clientes vem pela troca de tráfego de São Paulo, sendo assim, posso disponibilizar 40% a menos de trânsito IP nos meus link.

Com relação aos 500Mb de VPN sou obrigado a ter em SP a soma do total que tenho nas 5 terminações, ou seja, 100Mb de cada POP x 5=500Mb, porém uma boa negociação pode diminuir muito o preço do Mb de VPN, por exemplo, aqui na ANID pagamos menos da metade desse valor.

Vamos entender melhor:

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante



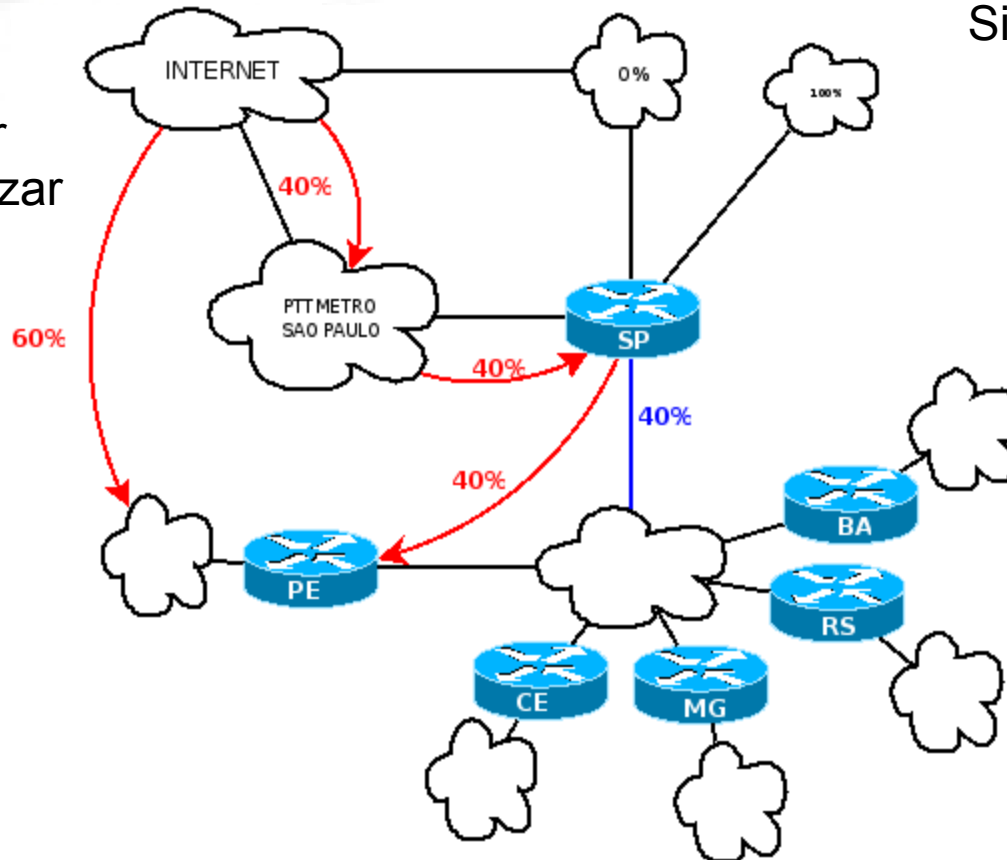
Situação Normal: 60% do tráfego chegando pelos links IP locais

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Situação Normal:

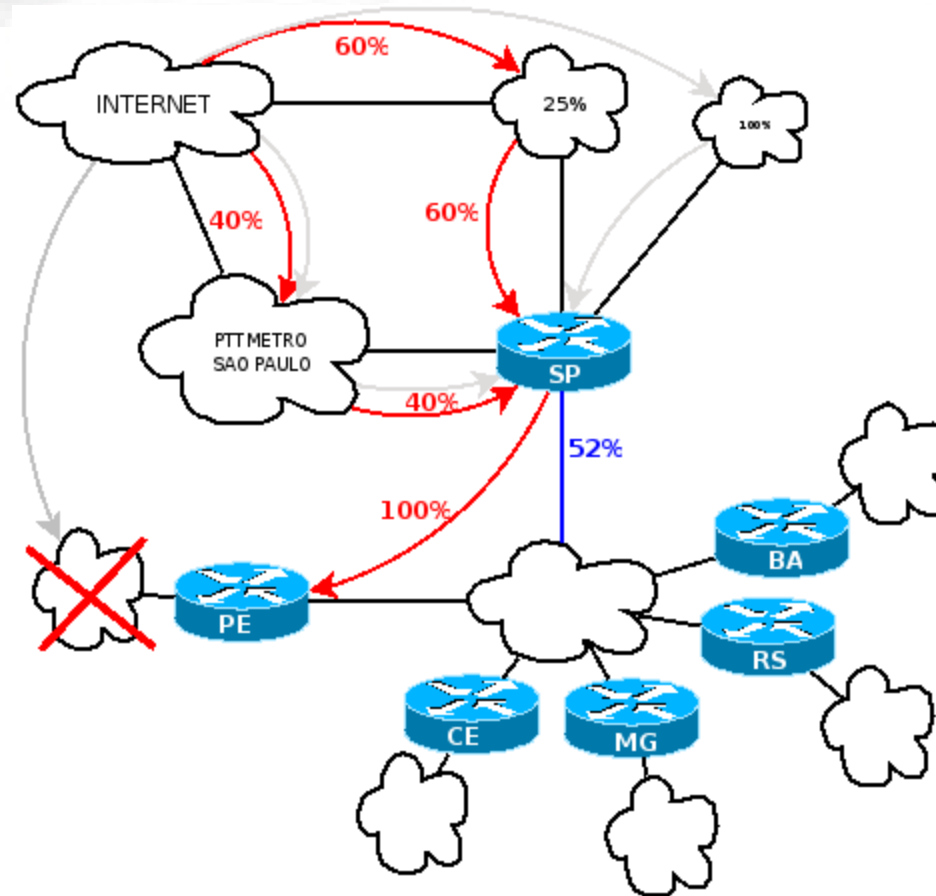
Para facilitar vamos analisar um POP por vez:



60% de todo tráfego chega pelo circuito IP local e 40% pelo link BGP-VPN

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante



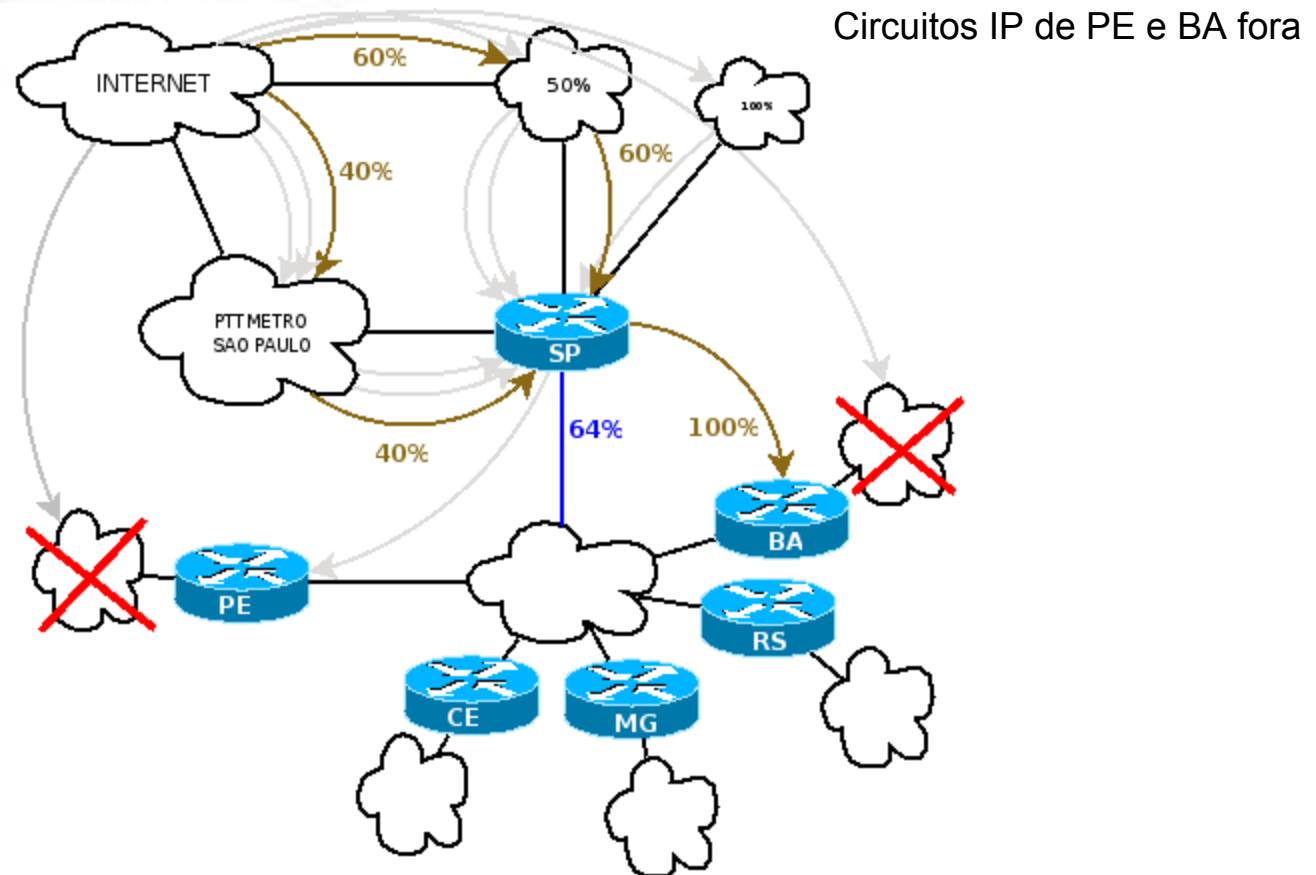
Circuito IP de PE fora

Agora 100% do me tráfego chega em PE pelo link BPG-VPN



## 5-Um estudo de caso

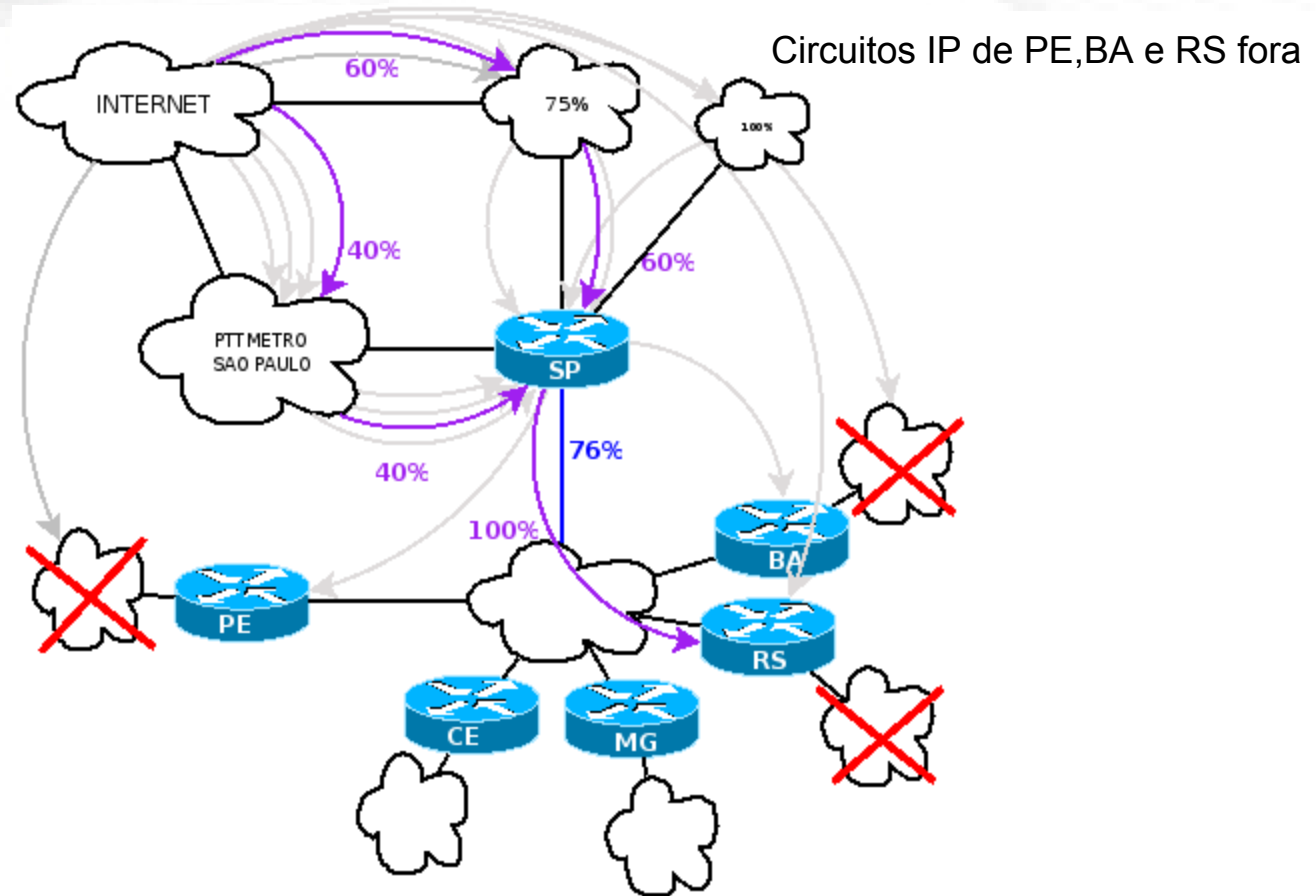
### 5.4 - Um modelo relativamente barato de link redundante



Ainda me restam 60Mb de banda IP aguardando o PIOR dos casos...

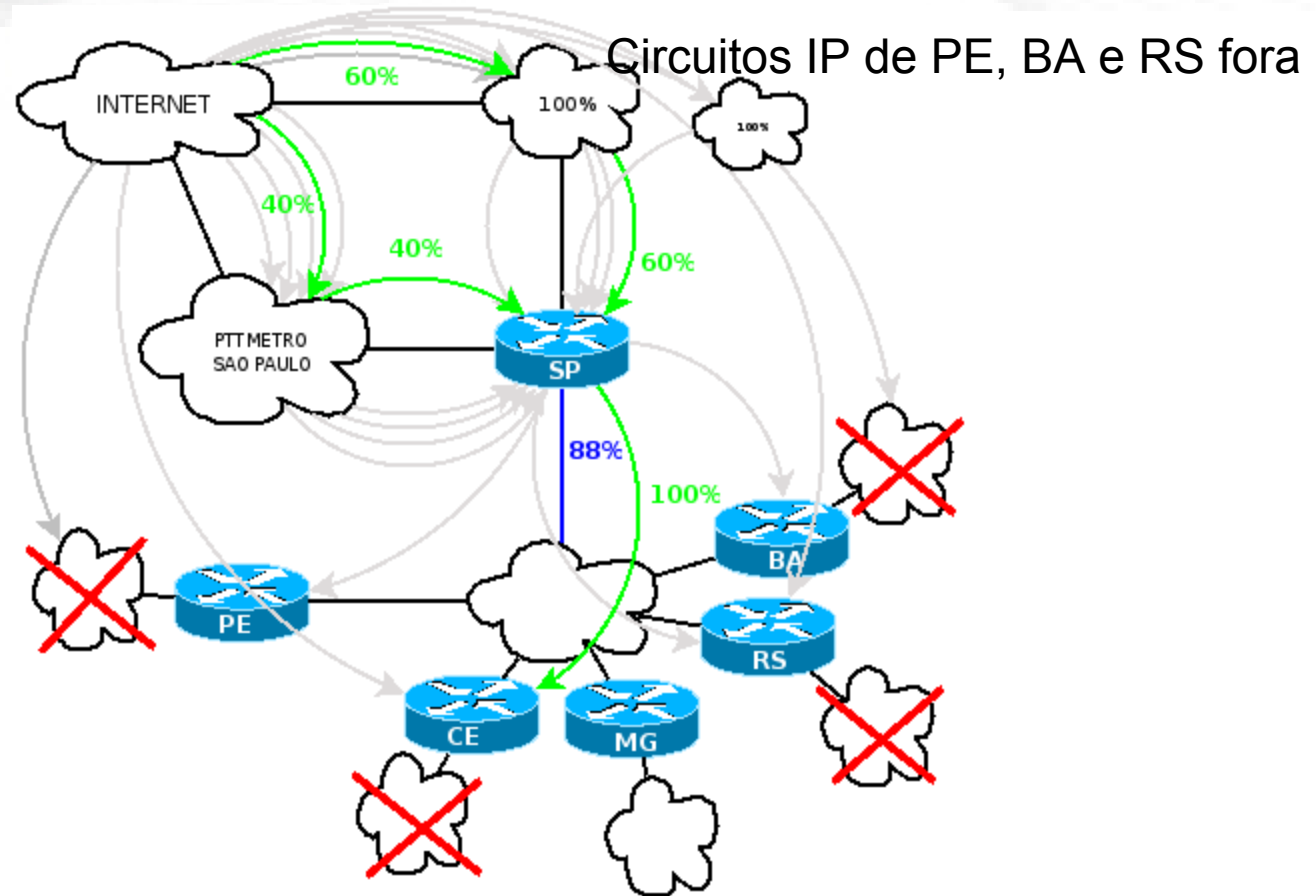
## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante



## 5-Um estudo de caso

## 5.4 - Um modelo relativamente barato de link redundante



## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando as sessões eBGP:

```
admin@SP] /routing bgp peer>
```

```
add name=link-SP-master remote-address=ip_do_peer_remoto  
remote-as=as_operadora in-filter=link-sp-master-in out-filter=link-sp-  
master-out nexthop-choice=force-self
```

```
add name=link-SP-backup remote-address=ip_do_peer_remoto  
remote-as=as_operadora in-filter=link-sp-backup-in out-filter=link-sp-  
backup-out nexthop-choice=force-self
```

```
add name=ptt-sp-rs1 remote-address=187.16.216.254 remote-  
as=26162 in-filter=ptt-sp-in out-filter=ptt-sp-out nexthop-  
choice=force-self
```

```
add name=ptt-sp-rs2 remote-address=187.16.216.253 remote-  
as=26162 in-filter=ptt-sp-in out-filter=ptt-sp-out nexthop-  
choice=force-self
```

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando as sessões iBGP:

```
admin@SP] /routing bgp peer>
```

```
add name=ibgp-PE remote-address=ip_do_pop_remoto remote-  
as=28135 in-filter=ibgp-in out-filter=ibgp-out route-reflect=yes default-  
originate=always nexthop-choice=force-self
```

```
add name=ibgp-CE remote-address=ip_do_pop_remoto remote-  
as=28135 in-filter=ibgp-in out-filter=ibgp-out route-reflect=yes default-  
originate=always nexthop-choice=force-self
```

```
add name=ibgp-MG remote-address=ip_do_pop_remoto remote-  
as=28135 in-filter=ibgp-in out-filter=ibgp-out route-reflect=yes default-  
originate=always nexthop-choice=force-self
```

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando as sessões IBGP:

```
admin@SP] /routing bgp peer>
```

```
add name=ibgp-RS remote-address=ip_do_pop_remoto remote-  
as=28135 in-filter=ibgp-in out-filter=ibgp-out route-reflect=yes default-  
originate=always nexthop-choice=force-self
```

```
add name=ibgp-BA remote-address=ip_do_pop_remoto remote-  
as=28135 in-filter=ibgp-in out-filter=ibgp-out route-reflect=yes default-  
originate=always nexthop-choice=force-self
```

Configurando iBGP no POP remoto (exemplo vale para todos)

```
admin@PE] /routing bgp peer>
```

```
add name=ibgp-SPO remote-address=ip_do_pop_remoto remote-  
as=28135 in-filter=ibgp-SPO-in out-filter=ibgp-SPO-out nexthop-  
choice=force-self
```

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal **ibgp-in**:

```
admin@SP] /routing filter>
```

```
add action=accept chain=ibgp-in set-bgp-communities=1111:1111  
set-bgp-local-pref=120
```

# Essa regra identifica com a community 1111:1111 e prioriza as rotas recebidas dos POPs remotos

\*quero que a prioridade dessas rotas seja maior que o padrão (100) recebido das operadoras, mas ao mesmo tempo menor do que as rotas recebidas do PTT

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal **ibgp-out**:

```
admin@SP] /routing filter>
```

```
add action=discard chain=ibgp-out set-bgp-as-path=^17379$
```

```
add action=accept chain=ibgp-out set-bgp-communities=999:999
```

# A primeira regra evita que rotas da nuvem BGP-VPN sejam enviadas para os POPs remotos pelo Route Reflector

# A segunda marca todas as rotas "refletidas" com a community 999:999



## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal **ptt-sp-in**:

```
admin@SP] /routing filter>
```

```
add action=accept chain=ptt-sp-in set-bgp-local-pref=110
```

# Essa regra prioriza as rotas recebidas do PTT-Metro em relação as rotas recebidas das operadoras, mas não em relação as rotas recebidas por IBGP que são meus clientes redes do meu AS

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal **ptt-sp-out**:

```
admin@SP] /routing filter>
```

```
add action=accept chain=ptt-sp-out bgp-communities=1111:1111  
add action=accept chain=ptt-sp-out prefix=redes_loais  
add action=discard chain=ptt-sp-out
```

# A primeira regra permite anúncio das redes recebidas por iBGP dos POPs remotos;

# A segunda permite o anúncio das redes do próprio POP que não receberam a community 1111:1111

# A terceira descarta todo o resto

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal **link-sp-master-in**:

```
admin@SP] /routing filter>
```

```
add action=accept chain=link-sp-master-in set-bgp-local-pref=90
```

# Essa regra atribui um Local\_preference de 90 para as rotas recebidas da operadora SP master com a finalidade de não permitir que os routers remotos utilizem essas rotas , mas sim as rotas recebidas das operadoras locais que terão Local\_pref padrão (100)

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal **link-sp-master-out**:

```
admin@SP] /routing filter>  
  
add action=discard chain=link-sp-master-out bgp-communities=1111:  
1111  
add action=accept chain=link-sp-master-out prefix=redes_locais  
add action=discard chain=link-sp-master-out
```

# A primeira regra evita o problema descrito na sessão 5.3

# A segunda permite o anúncio dos blocos de cliente locais

# A terceira descarta todo o resto e evita por exemplo o anúncio de redes recebidas do PTT-Metro

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal **link-sp-backup-in**:

```
admin@SP] /routing filter>
```

```
add action=accept chain=link-sp-backup-in set-bgp-local-pref=80
```

#Essa regra atribui um Local\_preference de 80 para as rotas recebidas do link backup para que as mesmas não subam na FIB do RouterSP, a menos é claro que o link master fique fora

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

Configurando os filtros do canal link-sp-backup-out:

Primeiramente é preciso ter em mente que o link BACKUP de São Paulo deve ficar em stand by, ou seja, não quero que nenhum tráfego dos POPs remotos ou redes locais cheguem por esse circuito com exceção de uma eventual queda nos circuitos IP principais (local ou qualquer um dos remotos)

Para isso iremos utilizar o AS-Patch-Prepend:

Vamos as configurações:

## 5-Um estudo de caso

### 5.4 - Um modelo relativamente barato de link redundante

# A primeira regra anuncia as rotas recebidas dos POPs remotos repetindo 3 vezes o meu próprio AS e remove a community 1111:1111

# A segunda faz a mesma coisa com as redes locais

# A Terceira não permite o anúncio de nada mais

```
admin@SP] /routing filter>
```

```
add action=accept chain=link-sp-backup-out  
set-bgp-prepend=3 bgp-communities=1111:1111 set-bgp-  
communities=""
```

```
add action=accept chain=link-sp-backup-out  
refix=redes_locais set-bgp-prepend=3
```

```
add action=discard chain=link-sp-backup-out
```

# PERGUNTAS??

peçoal: rinaldopvaz@gmail.com  
empresa rinaldo@anid.com.br

<http://rinaldovaz.blogspot.com>

@rinaldo\_vaz



# **AGRADECIMENTOS**

À todos que participam do projeto ANID, associados, mantenedores e conselheiros e colaboradores especialmente à equipe do NOC e Patrícia Correia pela valiosa contribuição na elaboração dessa apresentação!

## **OBRIGADO!**