

COMO FUNCIONA A INTERNET?

DESCOMPLICANDO O BORDER GATEWAY PROTOCOL (BGP)



SOBRE O PROFESSOR

Formação Acadêmica:

- (2022 - Atual): Doutorando em Ciência da Computação.
- (2014): Mestre em Ciência da Computação .
- (2018): Especialista em Sistemas de Informação.
- (2011): Bacharel em Engenharia de Computação.



Experiência Profissional:

- (2018 - Atual): Instrutor CCNA (Cisco Certified Network Associate) pela academia Cisco IFG – Campus Formosa –
- (2020 - Atual): Certificado pela Huawei Certified ICT Associate 5G e fundador da Academia Huawei no IFG – Campus Formosa.
- (2015 – Atual): Professor DE do IFG Campus Formosa com atuação nas seguintes áreas: Redes de Computadores, Sistemas Distribuídos, Segurança da Informação, Telecomunicações, Software/Hardware livre e Internet das Coisas.



GNS3

<https://www.gns3.com/>



Software Documentation Community Marketplace Academy



The software that empowers network professionals

Join the world's largest community of network professionals who rely on GNS3 to build better networks, share ideas and make connections.

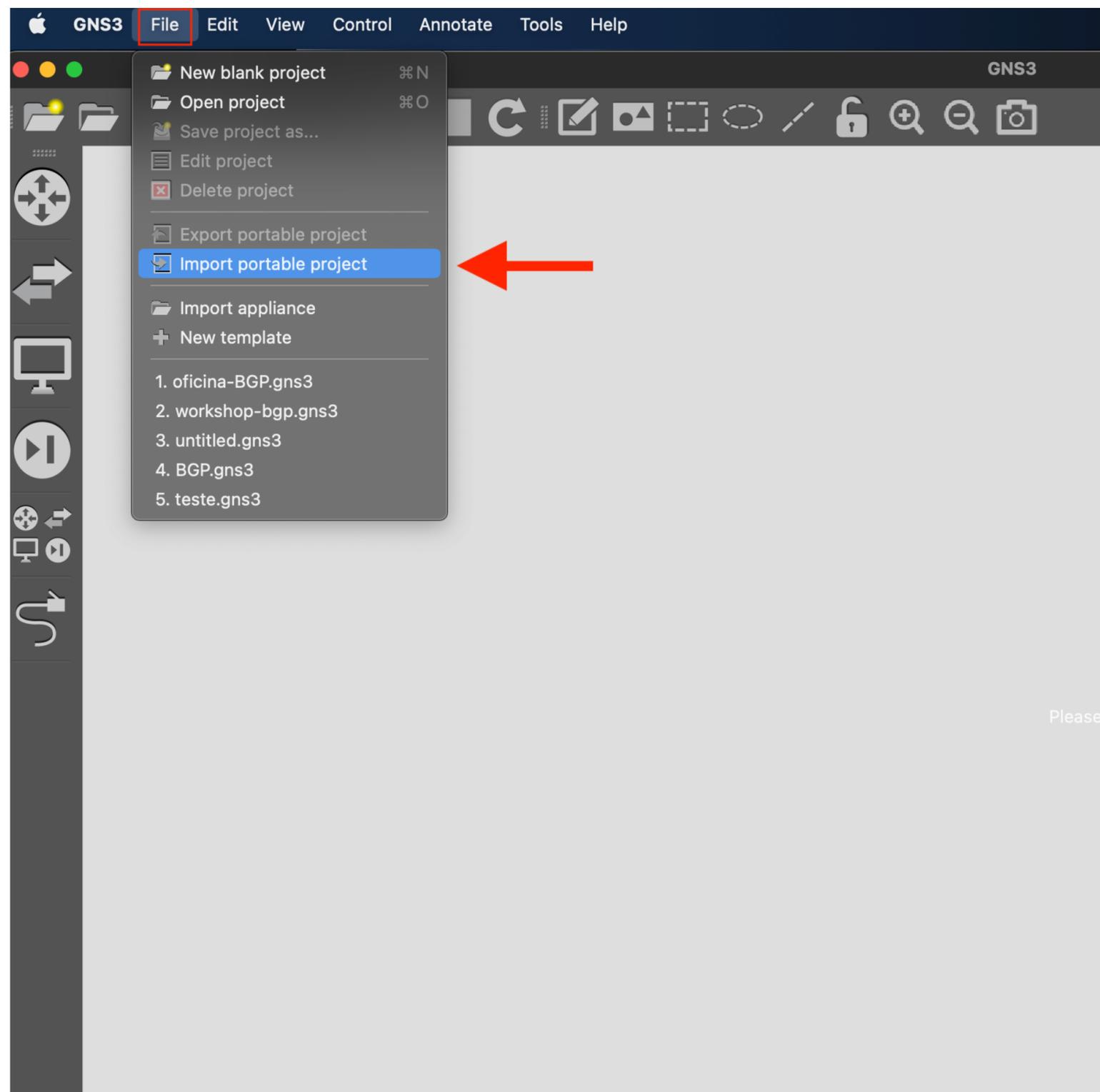
[Free Download](#) [Watch Video](#)

Download GNS3

Select the installer for your favourite OS

	Windows Version 2.2.35.1	Install Guide	Download
	Mac Version 2.2.35.1	Install Guide	Download
	Linux Version 2.2.35.1	Install Guide	Download

GNS3



Arquivo para ser importado



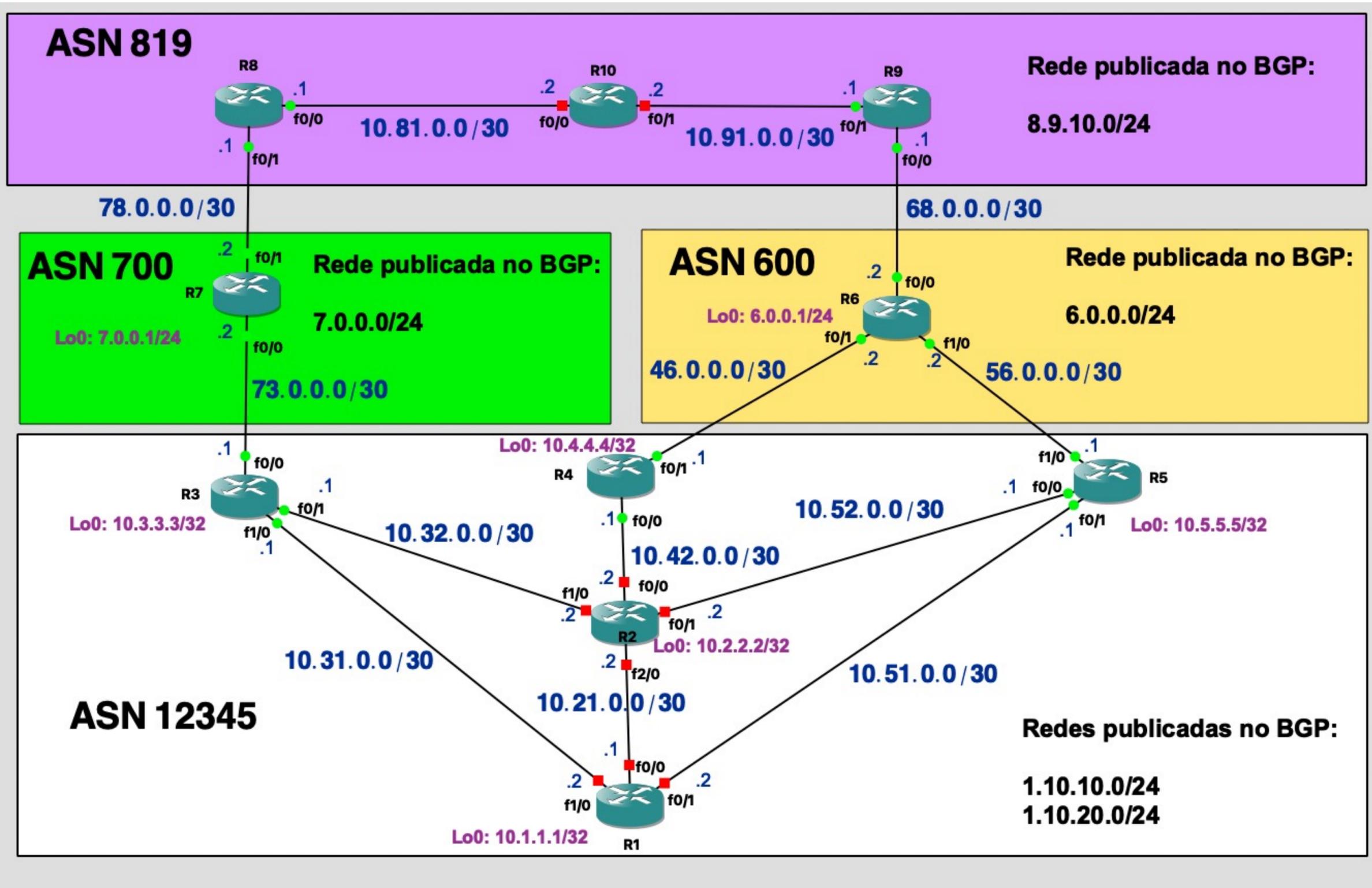
[oficina-BGP.gns3project](#)

(clique para fazer *download* ~ 170MB)

Snapshots:

- **inicial**: apenas o endereçamento de rede configurado.
- **config**: toda a configuração BGP deste minicurso

TOPOLOGIA – GNS3



ROTEAMENTO É O MESMO QUE ENCAMINHAMENTO ??

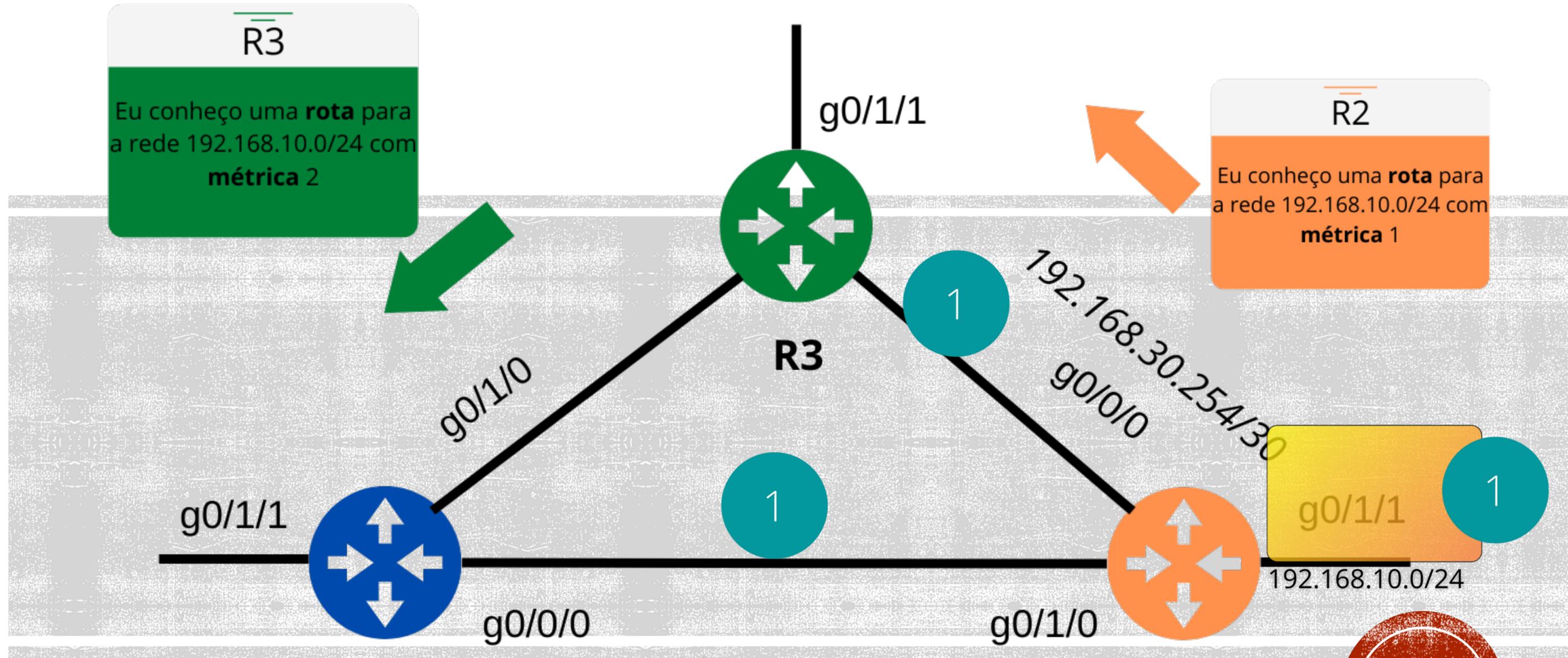
Roteamento é o processo que consiste em:

- i)** descobrir todas as possíveis rotas para cada uma das redes de destino,
- ii)** selecionar a melhor delas (com base em alguma métrica) e
- iii)** inserir esta melhor rota na Tabela de Rotas.

Quem faz o roteamento?:

- i)** administrador de rede (quando ele configura rotas estáticas)
- ii)** protocolos de roteamento (RIP, EIGRP, OSPF, **BGP**...).

Encaminhamento é o direcionamento de pacote para sua rede de destino de acordo com a Tabela de Rotas.



R1

Rotas aprendidas

Rede	Via	Métrica
192.168.10.0/24	R3	3
192.168.10.0/24	R2	2

R2

Eu conheço uma **rota** para a rede 192.168.10.0/24 com **métrica 1**

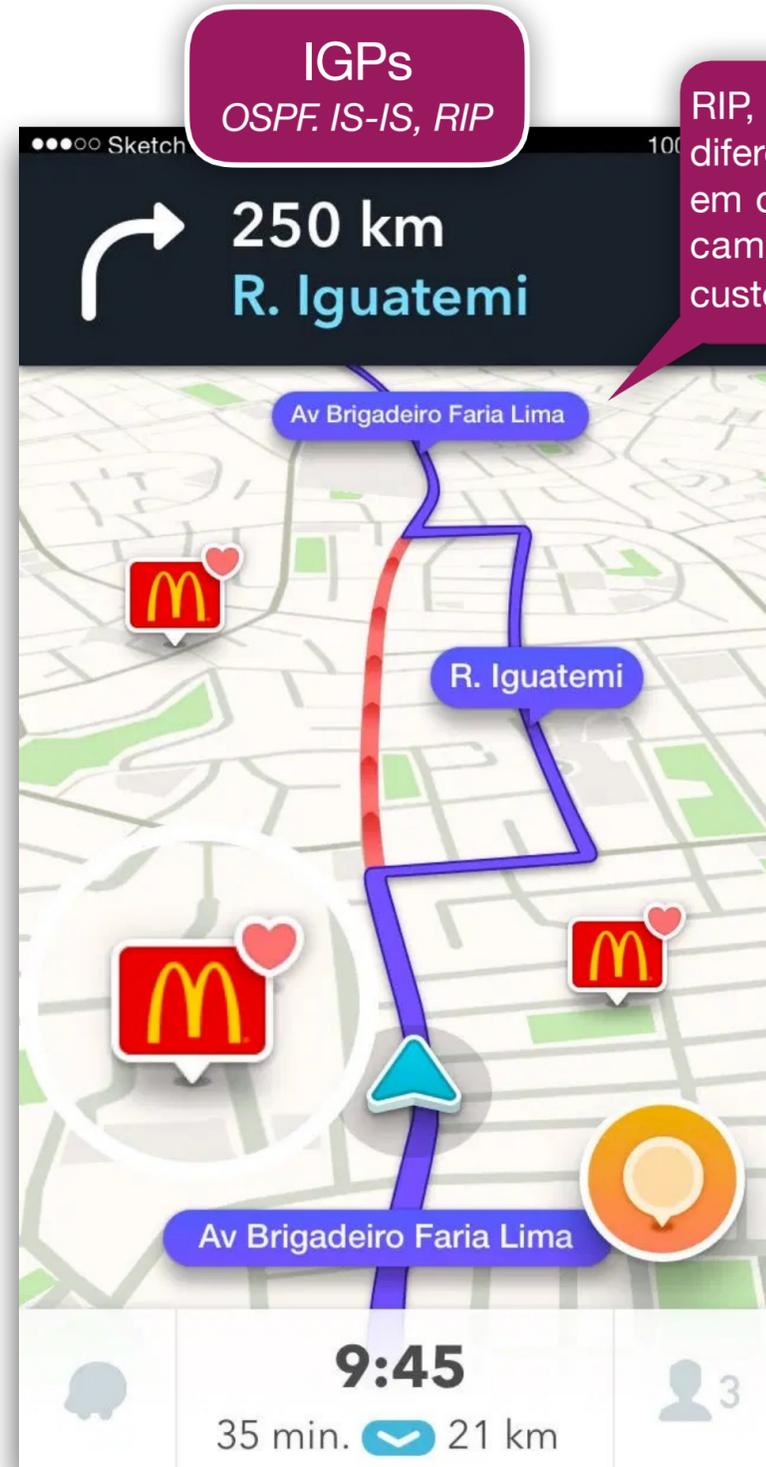
Subnet	Out. Int.	Next-Hop	Metric
192.168.10.0	g0/0/0	192.168.20.254	2

Tabela de Rotas

7

FUNDAMENTOS DO BGP

BGP (Border Gateway Protocol) é o protocolo de roteamento que é a **cola** utilizada para juntar as redes e transformar na Internet.



IGPs
OSPF, IS-IS, RIP

RIP, OSPF e EIGRP são todos diferentes, mas eles têm uma coisa em comum: eles querem encontrar o caminho mais curto (ou menos custoso) para o destino.

Quando olhamos para a Internet, **não nos importamos tanto em encontrar o caminho mais curto**, a possibilidade de **manipular** os caminhos do tráfego é muito mais importante.

Atualmente, existe apenas um protocolo de roteamento que usamos na Internet, o BGP.

Border Gateway Protocol (BGP) Versions		
RFC	DATA	VERSÃO
1105	1989	1
1163	1990	2
1267	1991	3
1654 e 1771	1994 e 1995	4

BGP
e Internet



BGP atualmente vai muito além do que "somente" o protocolo de roteamento da Internet, ele é amplamente utilizado em diversos cenários como DC, Cloud, Sec, IoT e outros.

SISTEMAS AUTÔNOMOS - AUTONOMOUS SYSTEMS

Para o roteamento entre os diferentes sistemas autônomos, usamos um EGP (protocolo de gateway externo). O único EGP que usamos atualmente é o BGP.

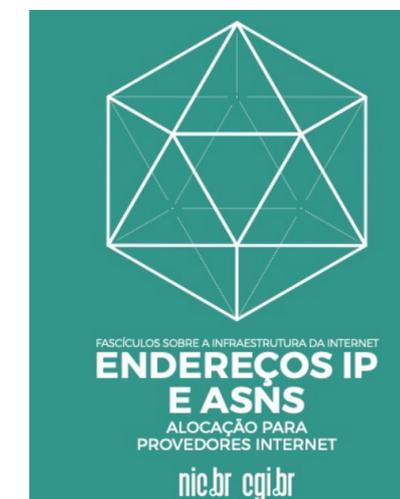
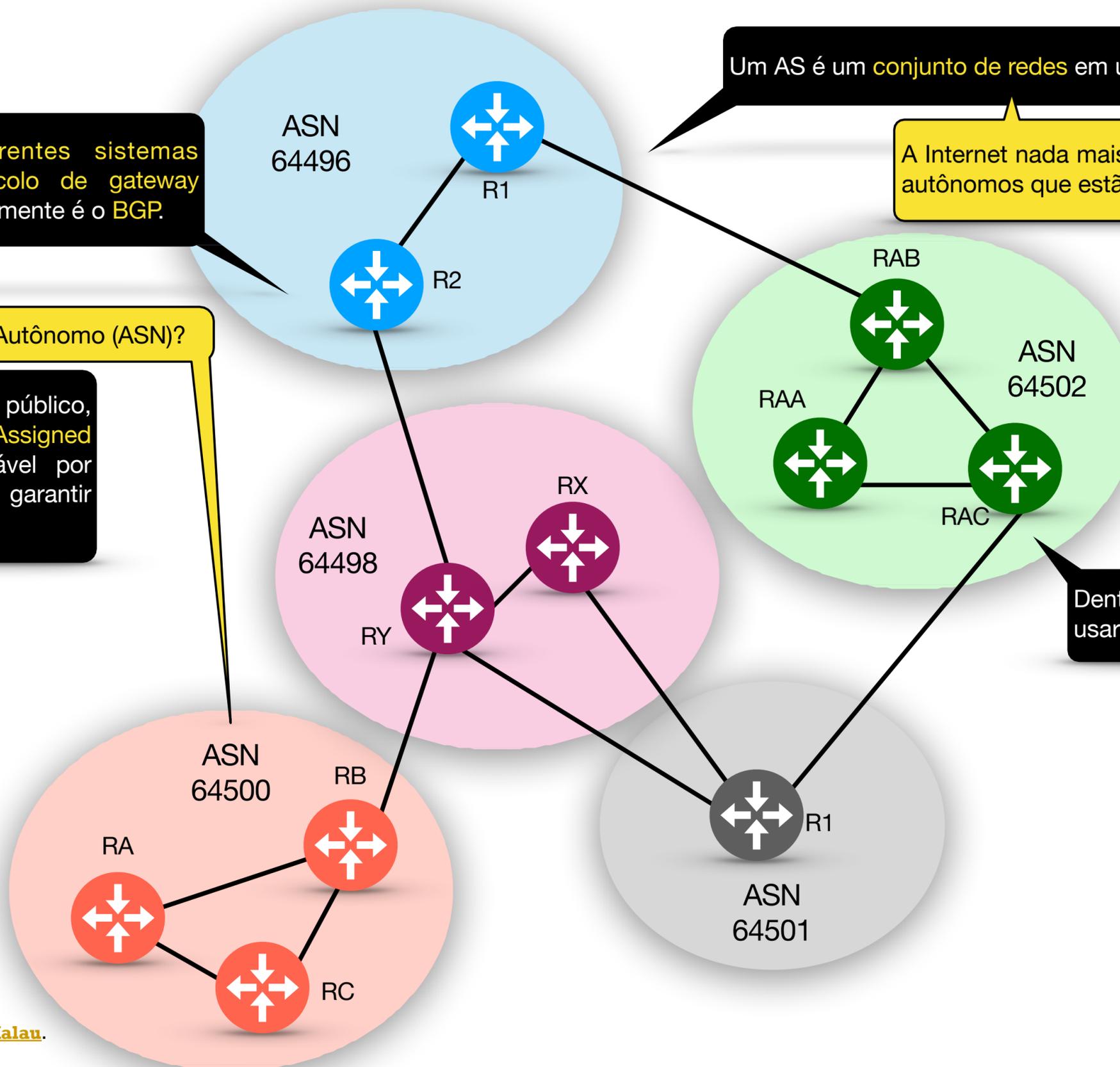
Como obtemos um Número de Sistema Autônomo (ASN)?

Assim como o range de endereço IP público, você precisará registrar um. A Internet Assigned Numbers Authority (IANA) é responsável por atribuir todos os ASNs públicos para garantir que eles sejam globalmente exclusivos.

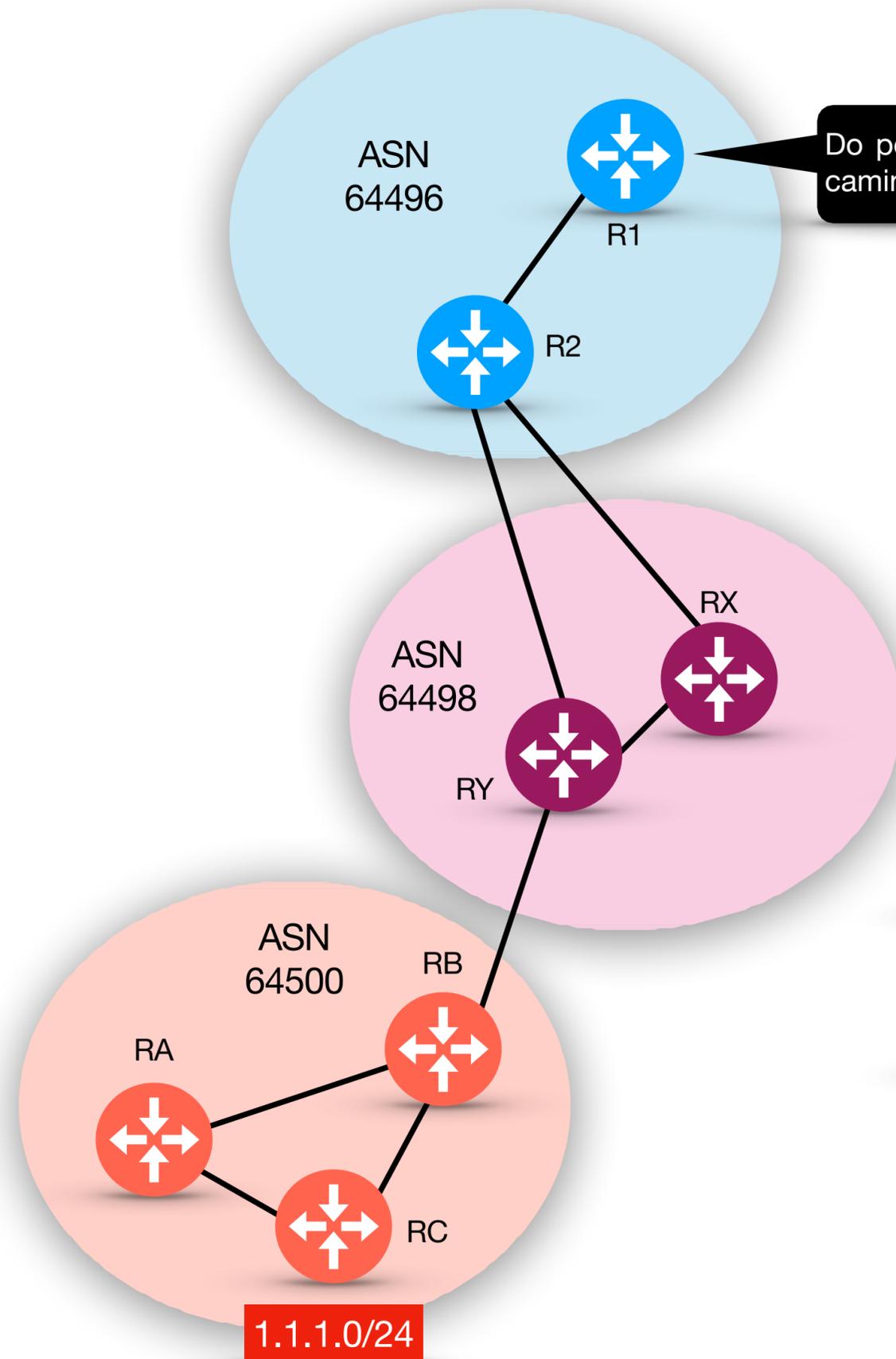
Um AS é um conjunto de redes em um único domínio administrativo.

A Internet nada mais é do que um monte de sistemas autônomos que estão conectados um ao outro.

Dentro de um sistema autônomo, podemos usar um IGP como o OSPF e IS-IS.



VETOR CAMINHO - PATH VECTOR - AS-PATH



Do ponto de vista do R1, para chegar na rede 1.1.1.0/24 ele tem o seguinte caminho, chamado AS-PATH: ASN 64496 -> ASN 64498 -> ASN 64500.

Para escolher melhor caminho (caso tenha mais de um caminho) para chegar na rede 1.1.1.0 o BGP analisa um conjunto de atributos chamados de Path Attributes ou PA.

BGP Path Attributes são classificados da seguinte maneira

Well-known (bem conhecidos): devem ser obrigatoriamente reconhecidos por todas as implementações de BGP.

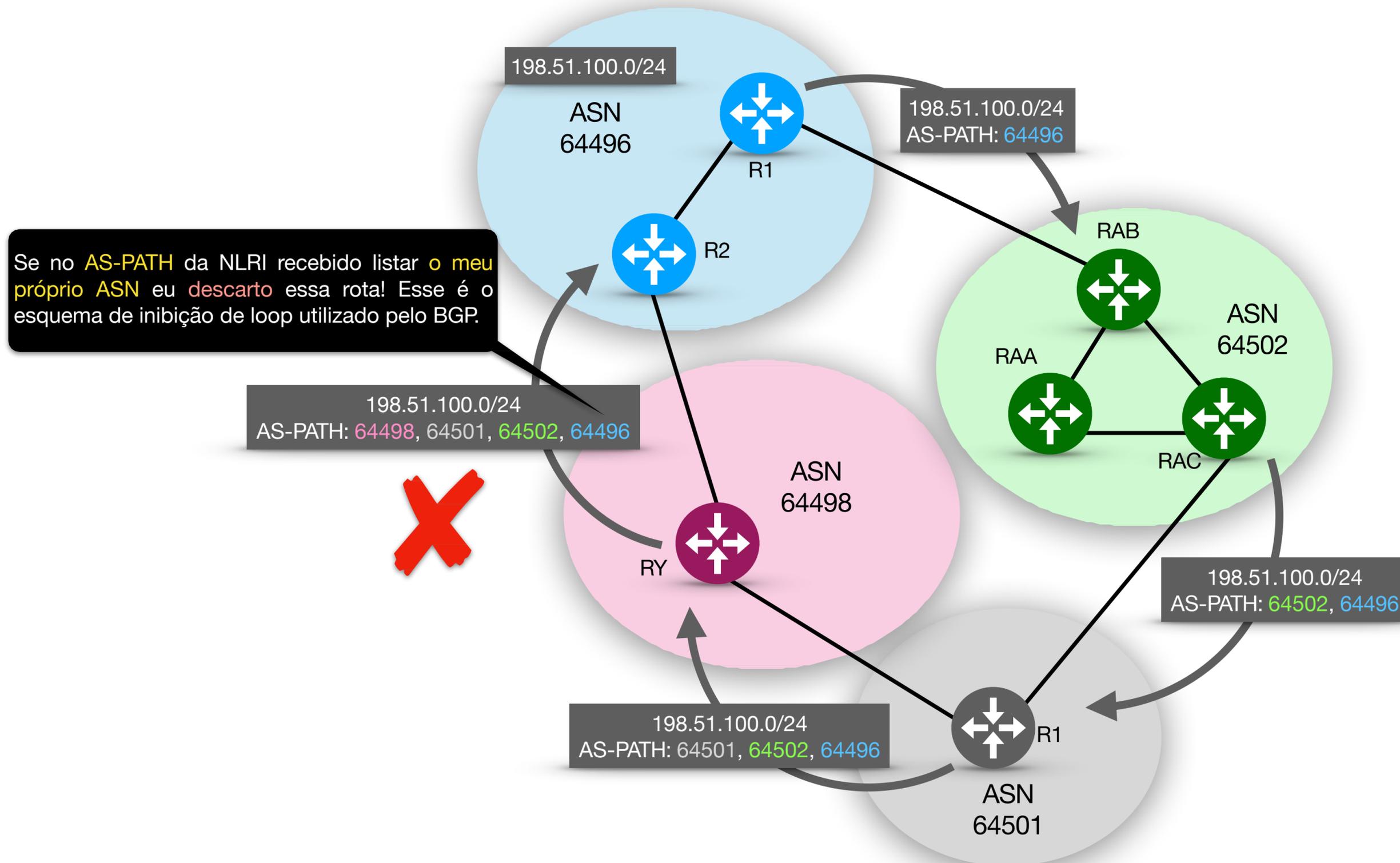
Optional: como o próprio nome diz, são atributos opcionais e sua implementação não é obrigatória.

BGP e o NLRI (Network Layer Reachability Information):

No BGP, o Network Layer Reachability Information (NLRI) é uma atualização de roteamento que consiste no prefixo da rede, no comprimento do prefixo e em quaisquer PAs (Path Attributes) BGP para a rota específica.

Ou seja, no BGP a divulgação do prefixo/redes contém muito mais informações que os protocolos de roteamento tradicionais, por isso não falamos que o BGP divulga uma rede e sim um NLRI (Network Layer Reachability Information).

PREVENÇÃO DE LOOP - BGP LOOP PREVENTION



INTER-ROUTER COMMUNICATION

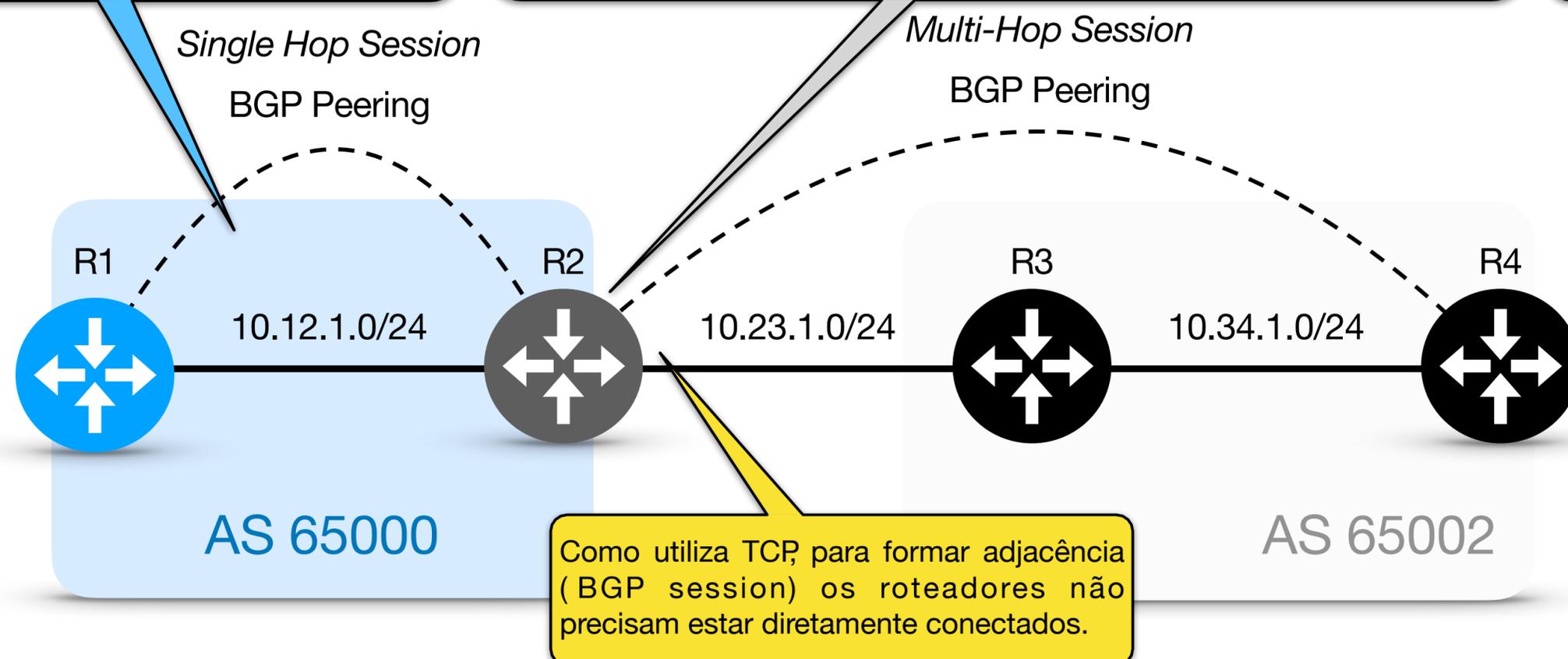
BGP não utiliza pacotes hello para descobrir vizinhos e a vizinhança não pode ser formada dinamicamente.

Ele utiliza TCP porta 179 para se comunicar com outros roteadores BGP.

Internal BGP (iBGP): Sessão BGP formada entre roteadores do mesmo AS, prefixos são colocados com administrative distance (AD) 200. TTL nos pacotes iBGP são setados como 255.

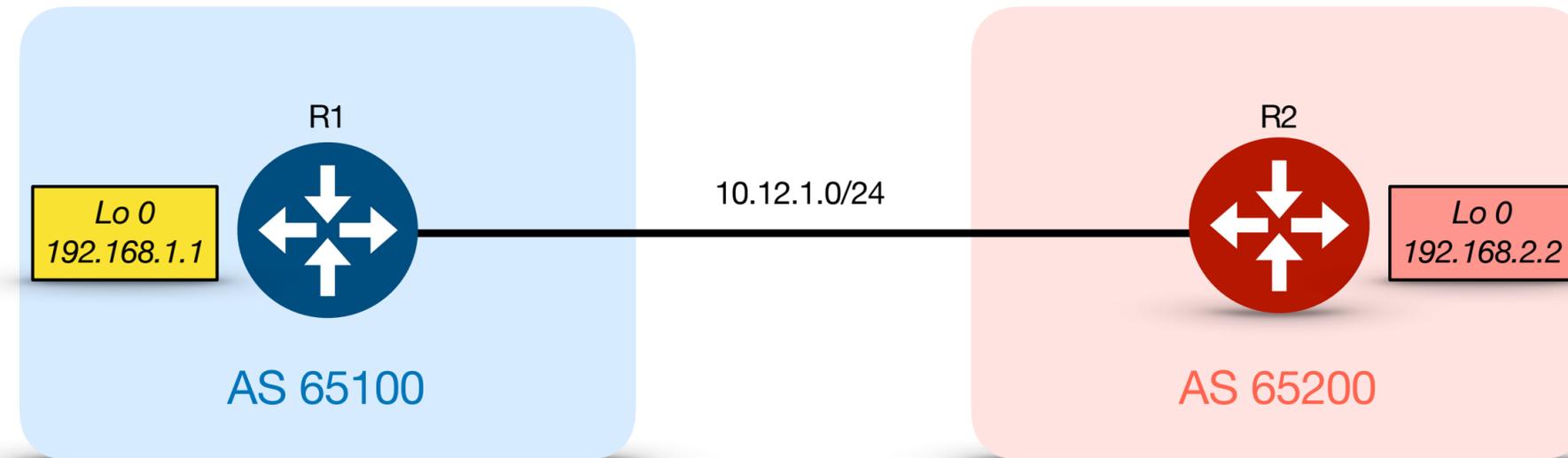
External BGP (eBGP): Sessão BGP formada entre roteadores de diferentes AS, prefixos são colocados com administrative distance (AD) 20. TTL nos pacotes eBGP são setados como 1, então por padrão a Multi-Hop session não pode ser feita no eBGP (esse comportamento pode ser alterado).

No eBGP as rotas enviadas para um vizinho tem o IP do próximo salto alterado para o IP do próprio roteador que envia as rotas, coisa que não acontece no iBGP.



Nesse caso por exemplo, o R2 precisa ter uma rota para alcançar o R4 e o R4 deve alcançar o R2, pode ser uma rota estática ou um IGP como o OSPF, **rota default não funciona.**

VERIFICATION OF BGP SESSIONS



Existe também o comando `show ip bgp summary` (foi criado antes do MPBGP).

Período de tempo em que a sessão BGP é estabelecida ou o status atual se a sessão não estiver estabelecida.

```
R1# show bgp ipv4 unicast summary
BGP router identifier 192.168.2.2, local AS number 65200
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.12.1.2	4	65200	8	7	3	0	0	00:05:23	2

Estado atual do peer BGP ou o número de prefixos recebidos do peer

Endereço IP do peer BGP

Número do sistema autônomo do peer BGP

Contagem de mensagens enviadas para o peer BGP

Número de mensagens enfileiradas para serem processadas pelo peer

Versão BGP falada pelo peer BGP

Contagem de mensagens recebidas do peer BGP

Última versão da tabela BGP enviada ao peer

Número de mensagens enfileiradas para serem enviadas ao peer

BGP FINITE-STATE MACHINE - FSM

Idle (Ocioso) é o primeiro estágio do BGP FSM. O BGP detecta um evento inicial e tenta iniciar uma conexão TCP com o peer BGP e também escuta uma nova conexão a partir de um peer.

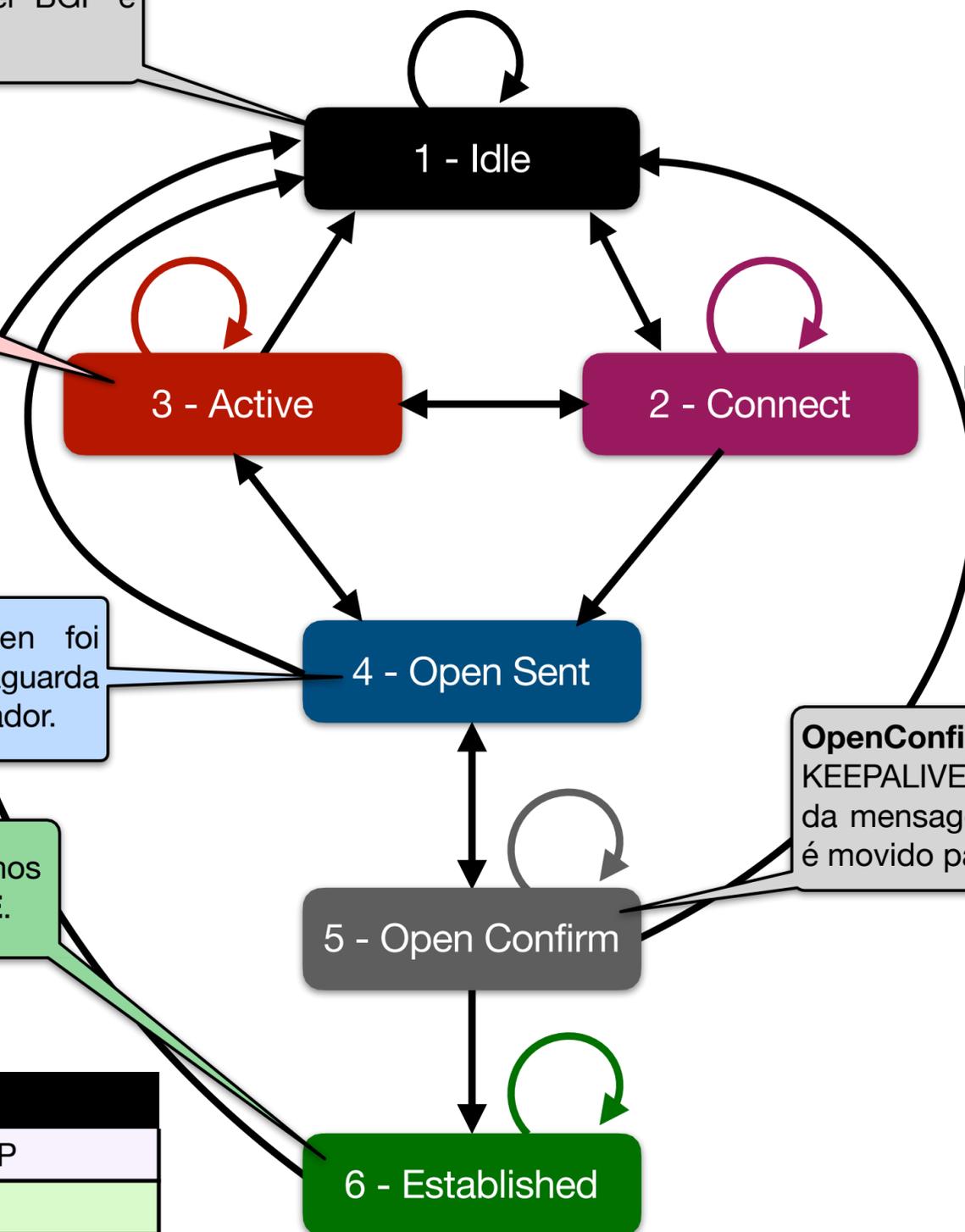
No **Active state**, o BGP inicia um novo three-way handshake TCP.

No estado **Connect**, o BGP inicia a conexão TCP.

OpenSent, uma mensagem Open foi enviada do roteador de origem e aguarda uma mensagem Open do outro roteador.

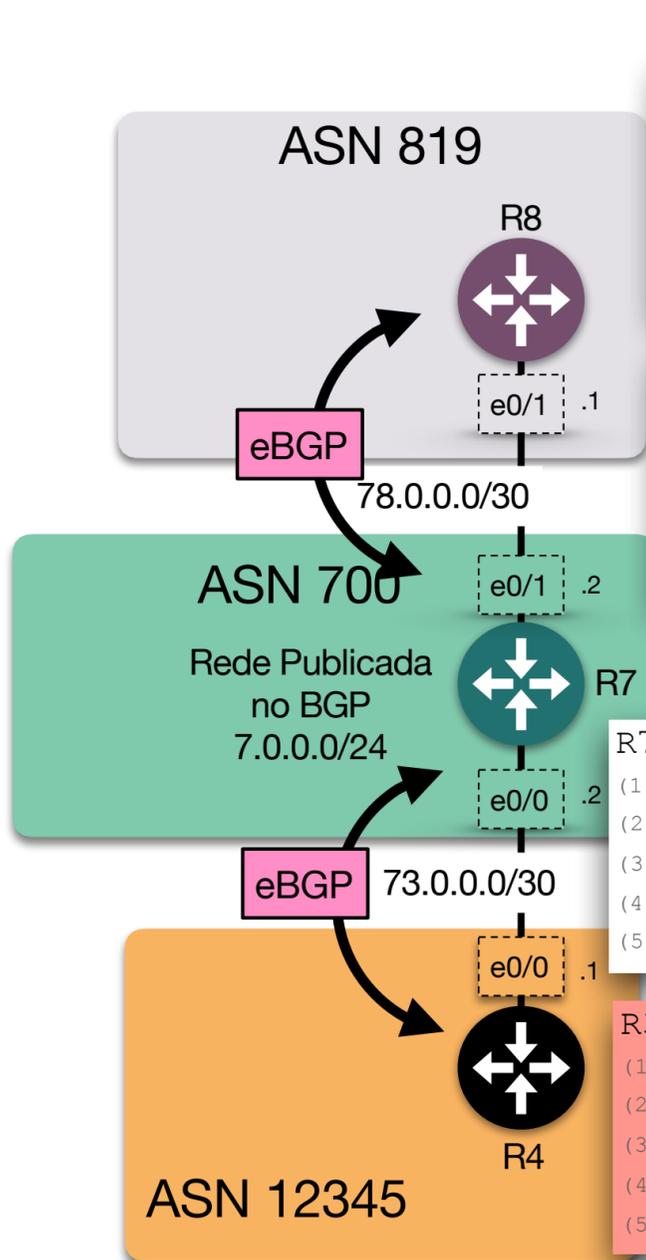
OpenConfirm, o BGP aguarda uma mensagem KEEPALIVE ou NOTIFICATION. Após o recebimento da mensagem KEEPALIVE de um vizinho, o estado é movido para Established.

Established, a sessão BGP é estabelecida. Os vizinhos BGP trocam NLRI/rotas usando mensagens **UPDATE**.



Tipo	Nome	Overview
1	OPEN	Configura e estabelece a adjacência BGP
2	UPDATE	Anuncia, atualiza ou retira rotas
3	NOTIFICATION	Indica uma condição de erro para um vizinho BGP
4	KEEPALIVE	Garante que os vizinhos BGP ainda estejam vivos

CONFIGURAÇÃO PARTE 1 - CONFIGURAR EBGP NO R8, R7 E R3



1

```
R8
(1) conf t
(2) router bgp 819
(3) neighbor 78.0.0.2 remote-as 700
(4) end
(5) wr
```

ou pode usar o: sh bgp ipv4 unicast summary

```
R8#sh ip bgp summary
BGP router identifier 78.0.0.1, local AS number 819
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
78.0.0.2	4	700	0	0	1	0	0	never	Idle

2

```
R7
(1) conf t
(2) router bgp 700
(3) neighbor 78.0.0.1 remote-as 819
(4) end
(5) wr
```

```
R7#sh ip bgp summary
BGP router identifier 7.0.0.1, local AS number 700
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
78.0.0.1	4	819	4	4	1	0	0	00:00:44	0

3

```
R7
(1) conf t
(2) router bgp 700
(3) neighbor 73.0.0.1 remote-as 12345
(4) end
(5) wr
```

```
R7#sh ip bgp summary
BGP router identifier 7.0.0.1, local AS number 700
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
73.0.0.1	4	12345	0	0	1	0	0	never	Idle
78.0.0.1	4	819	6	6	1	0	0	00:02:22	0

4

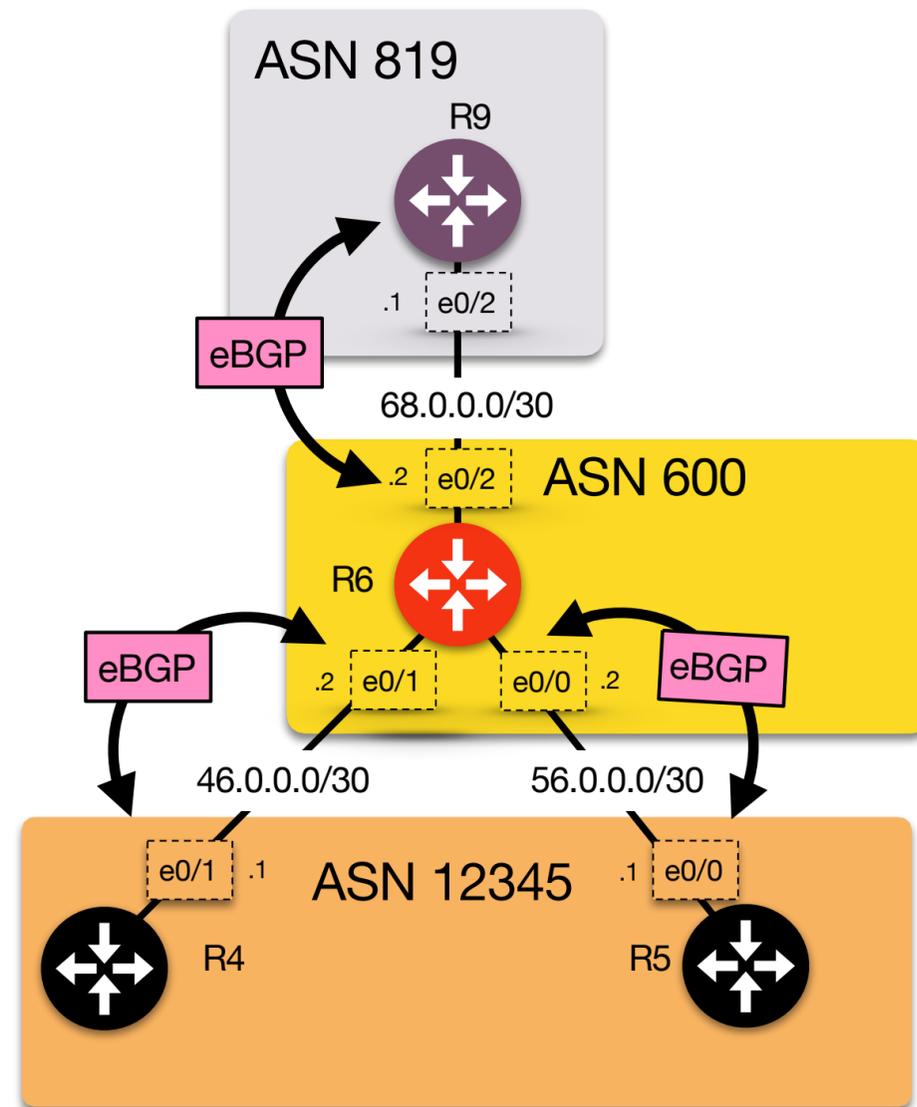
```
R3 - Simulando um erro:
(1) R3#conf t
(2) R3(config)#router bgp 12345
(3) R3(config-router)#neighbor 73.0.0.2 remote-as 770
(4) R3(config-router)#end
(5) R3#wr
```

```
R3#
*Jul  6 19:13:56.882: %BGP-4-MSGDUMP: unsupported or mal-
formatted message received from 73.0.0.2:
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 0039 0104 02BC 00B4
0700 0001 1C02 0601
0400 0100 0102 0280 0002 0202 0002 0246 0002 0641 0400 0002
BC
```

5

```
R3
(1) conf t
(2) no logging console
(3) router bgp 12345
(4) no neighbor 73.0.0.2 remote-as 770
(5) neighbor 73.0.0.2 remote-as 700
(6) exit
(7) logging console
(8) end
(9) wr
```

CONFIGURAÇÃO PARTE 2 - CONFIGURAR EBGP NO R9, R6, R4 E R5



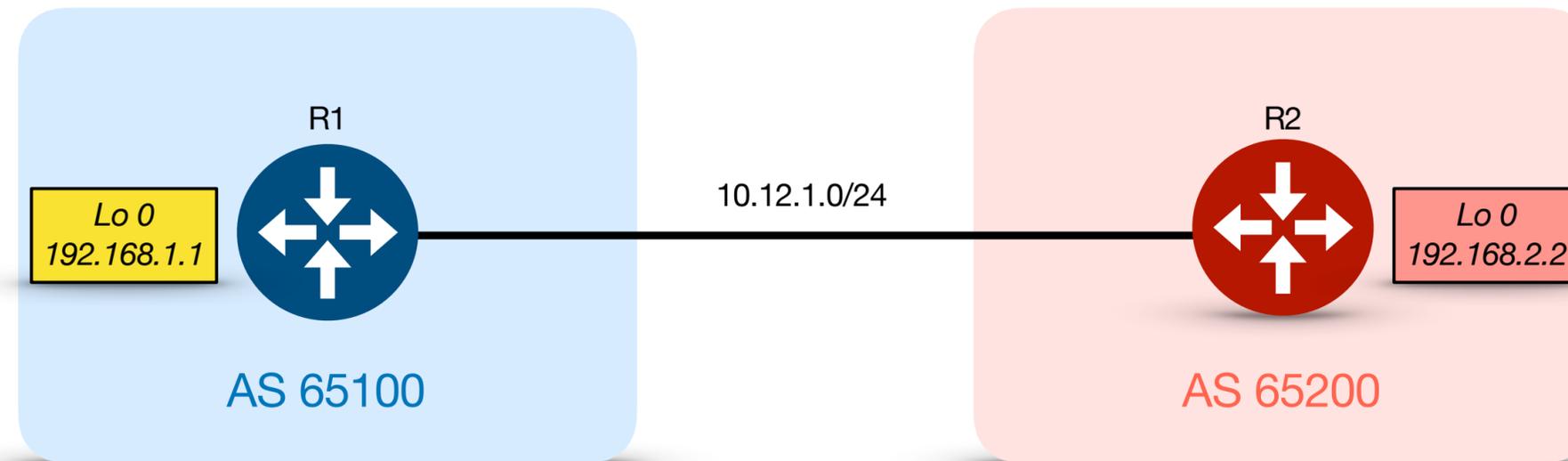
```
R9
(1) conf t
(2) router bgp 819
(3) neighbor 68.0.0.2 remote-as 600
(4) end
(5) wr
```

```
R6
(1) conf t
(2) router bgp 600
(3) neighbor 68.0.0.1 remote-as 819
(4) neighbor 46.0.0.1 remote-as 12345
(5) neighbor 56.0.0.1 remote-as 12345
(6) end
(7) R6#wr
```

```
R4
(1) conf t
(2) router bgp 12345
(3) neighbor 46.0.0.2 remote-as 600
(4) end
(5) wr
```

```
R5
(1) conf t
(2) router bgp 12345
(3) neighbor 56.0.0.2 remote-as 600
(4) end
(5) wr
```

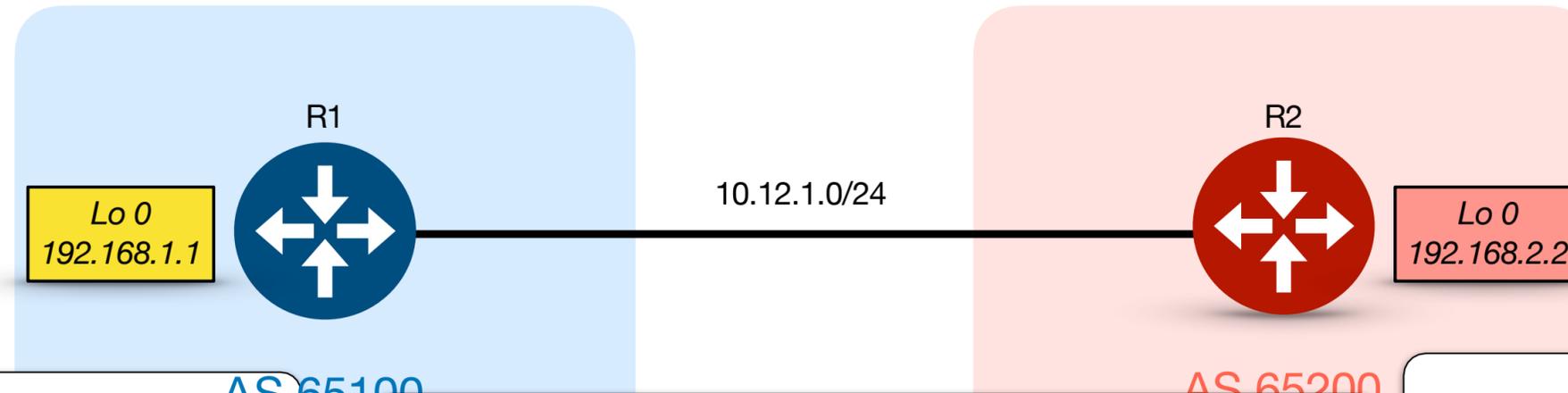
DIVULGANDO PREFIXOS (REDES)



```
R1
router bgp 65100
  bgp log-neighbor-changes
  neighbor 10.12.1.2 remote-as 65200
  network 10.12.1.0 mask 255.255.255.0
  network 192.168.1.1 mask 255.255.255.255
```

```
R2
router bgp 65200
  bgp log-neighbor-changes
  no bgp default ipv4-unicast
  neighbor 10.12.1.1 remote-as 65100
  !
  address-family ipv4
    network 10.12.1.0 mask 255.255.255.0
    network 192.168.2.2 mask 255.255.255.255
  neighbor 10.12.1.1 activate
  exit-address-family
```

DIVULGANDO PREFIXOS (REDES)



Network: Uma lista dos prefixos de rede instalados no BGP. Se existirem vários NLRI para o mesmo prefixo, apenas o primeiro prefixo será identificado e os outros ficarão em branco.

LocPref: Well-known discretionary BGP path attribute usado no algoritmo de melhor caminho BGP para o NLRI específico.

Weight: Um atributo definido pela Cisco, tem significância local e é usado no algoritmo de melhor caminho BGP para o NLRI específico.

```
R1# show bgp ipv4 unicast
BGP table version is 4, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

Network          Next Hop        Metric LocPrf  Weight Path
* 10.12.1.0/24    10.12.1.2       0         0 65200 i
*>
*> 192.168.1.1/32 0.0.0.0         0        32768 i
*> 192.168.2.2/32 10.12.1.2       0         0 65200 i
```

Os NLRI's válidos são indicados pelo *.

O NLRI selecionado como o melhor caminho é indicado por um >.

Path: well-known mandatory BGP path attribute usado para prevenção de loop e no algoritmo de melhor caminho BGP para o NLRI específico.

Next Hop - Um well-known mandatory BGP path attribute que define o endereço IP para o próximo salto para esse NLRI específico.

Multiple-exit discriminator (MED): Optional non-transitive BGP path attribute usado no BGP para o NLRI específico.

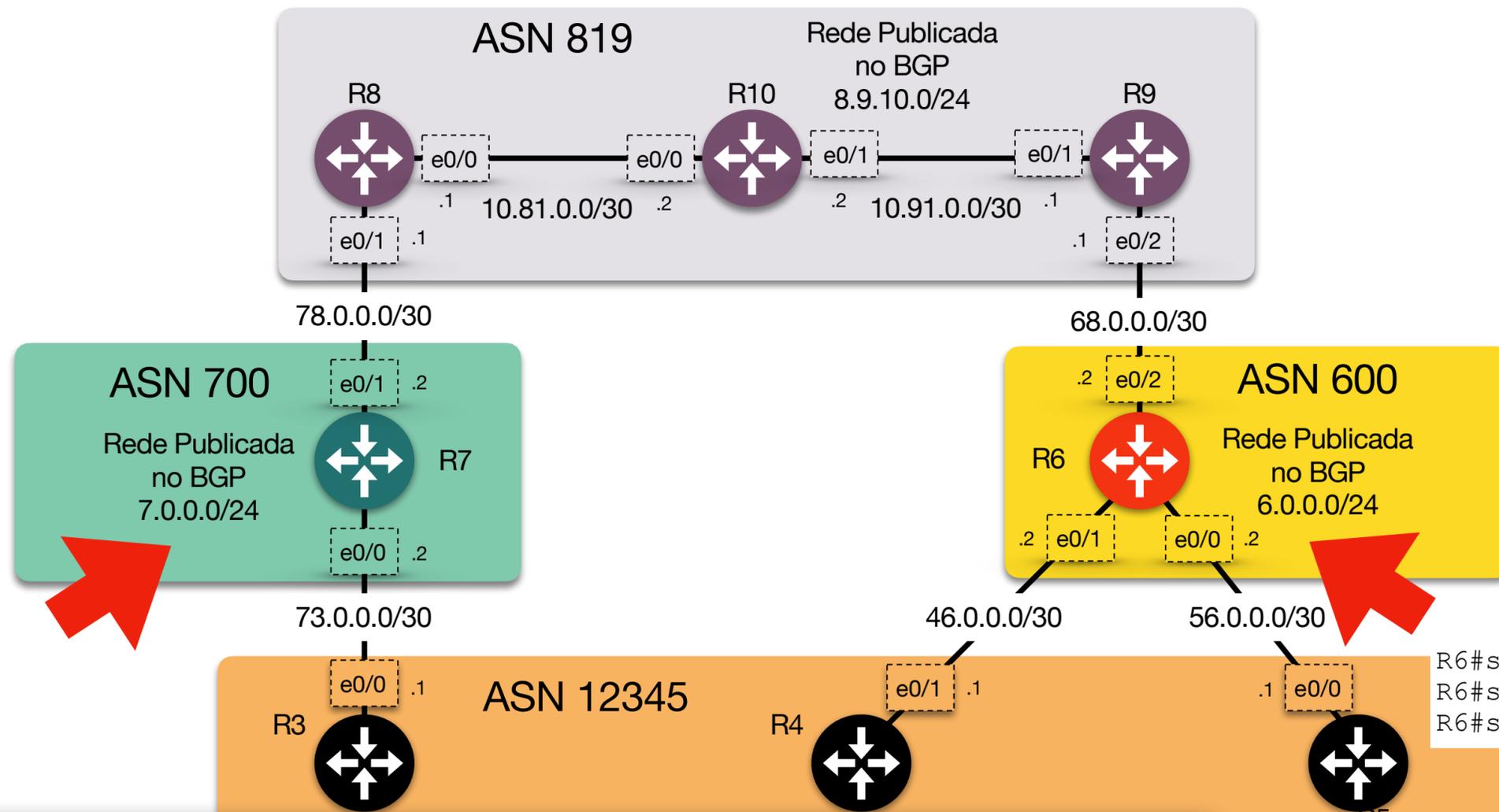
Origin: Well-known mandatory BGP path attribute usado no algoritmo de melhor caminho BGP.

```
R2# show bgp ipv4 unicast
BGP table version is 4, local router ID is 192.168.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

Network          Next Hop        Metric LocPrf  Weight Path
* 10.12.1.0/24    10.12.1.1       0         0 65100 i
*>
*> 192.168.1.1/32 10.12.1.1       0         0 65100 i
*> 192.168.2.2/32 0.0.0.0         0        32768 i
```

Um valor "i" representa um IGP, o "e" indica EGP e o "?" indica uma rota que foi redistribuída em BGP

CONFIGURAÇÃO PARTE 3 - DIVULGAR PREFIXOS DO R7 E R6 E VERIFICAR AS ROTAS



1

```
R7
(1) conf t
(2) router bgp 700
(3) network 7.0.0.0 mask 255.255.255.0
(4) end
(5) wr
```

2

```
R6
(1) conf t
(2) router bgp 600
(3) network 6.0.0.0 mask 255.255.255.0
(4) end
(5) wr
```

```
R6#sh ip bgp ipv4 unicast neighbors 46.0.0.1 advertised-routes
R6#sh ip bgp ipv4 unicast neighbors 56.0.0.1 advertised-routes
R6#sh ip bgp ipv4 unicast neighbors 68.0.0.1 advertised-routes
```

```
R7#sh bgp ipv4 unicast neighbors 78.0.0.1 advertised-routes
BGP table version is 2, local router ID is 7.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

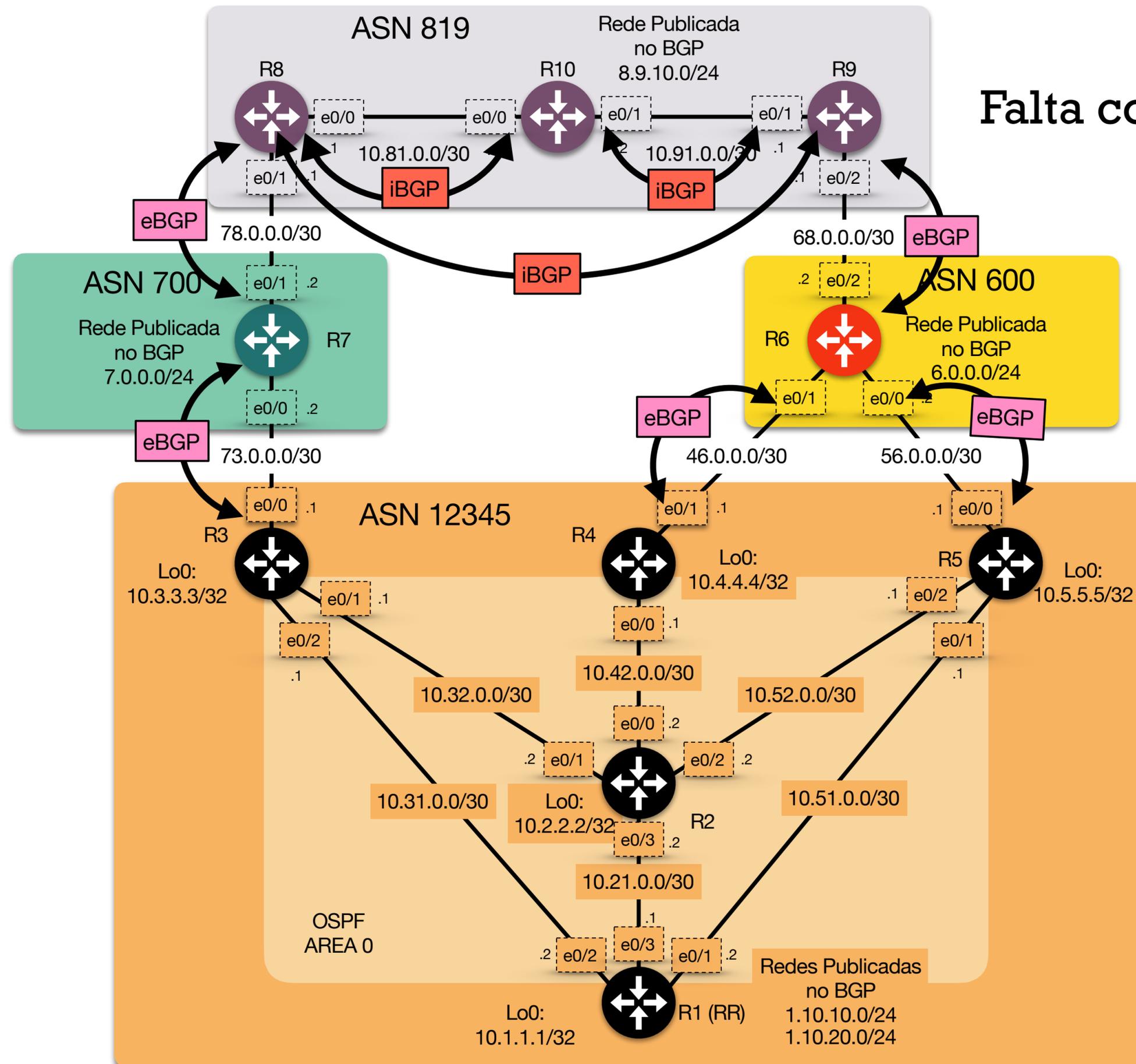
   Network          Next Hop          Metric LocPrf Weight Path
* >  7.0.0.0/24      0.0.0.0              0           32768 i

Total number of prefixes 1
```

```
R3#sh ip bgp
BGP table version is 2, local router ID is 10.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
* >  7.0.0.0/24      73.0.0.2              0           0 700 i
```

Falta configurar o iBGP!



APRENDIZAGEM DE PREFIXOS EBGP X IBGP

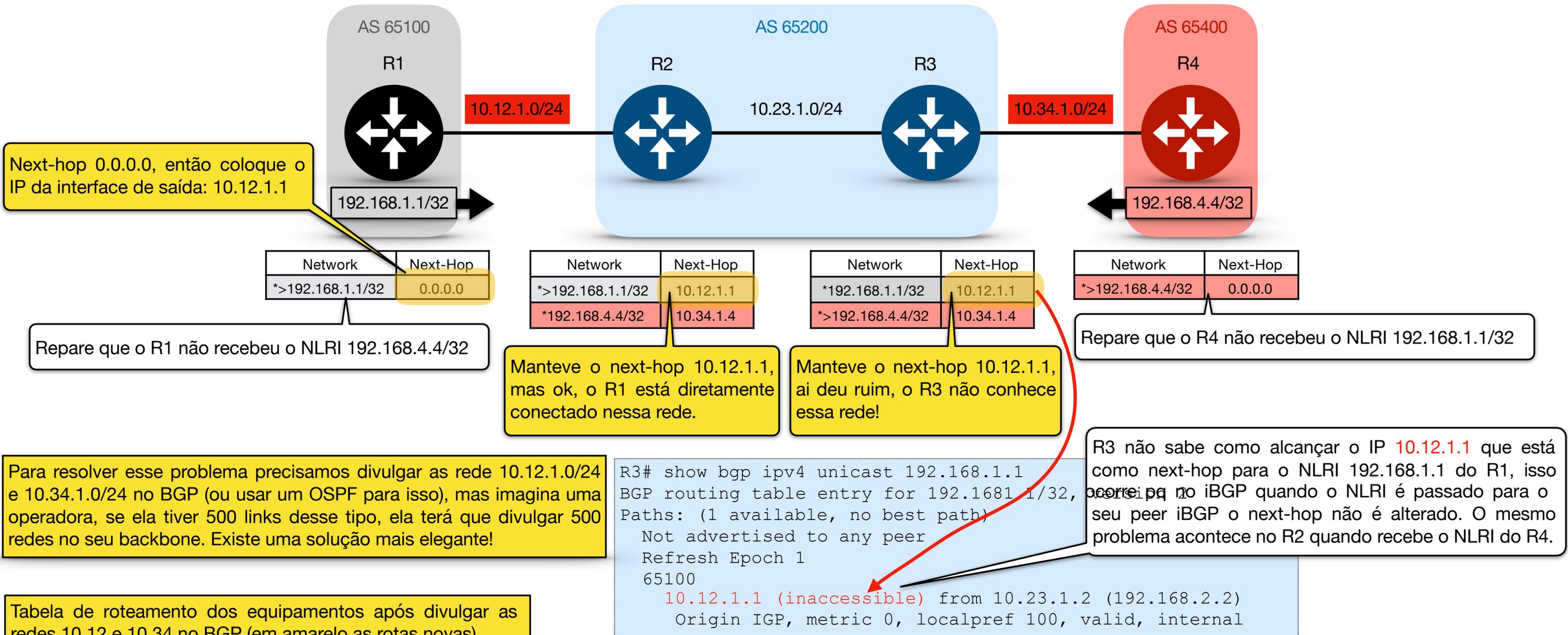
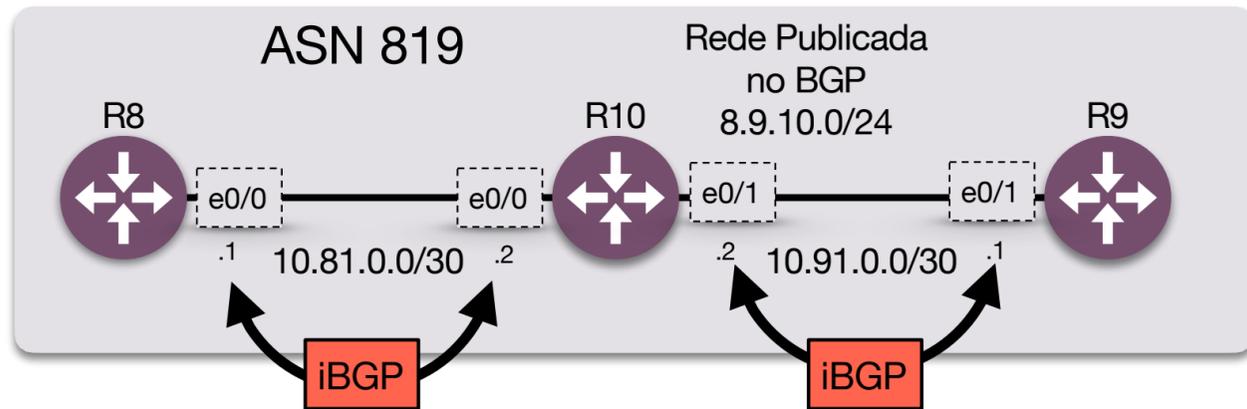


Tabela de roteamento dos equipamentos após divulgar as redes 10.12 e 10.34 no BGP (em amarelo as rotas novas).

R1		R2		R3		R4	
Network	Next-Hop	Network	Next-Hop	Network	Next-Hop	Network	Next-Hop
*>192.168.1.1/32	0.0.0.0	*>192.168.1.1/32	10.12.1.1	*>192.168.1.1/32	10.12.1.1	*>192.168.1.1/32	10.12.1.1
*>192.168.4.4/32	10.34.1.4	*>192.168.4.4/32	10.34.1.4	*>192.168.4.4/32	10.34.1.4	*>192.168.4.4/32	0.0.0.0
*>10.12.1.0/24	0.0.0.0	*>10.12.1.0/24	0.0.0.0	*>10.12.1.0/24	10.23.1.2	*>10.12.1.0/24	10.34.1.3
*>10.34.1.0/24	10.12.1.2	*>10.34.1.0/24	10.23.1.3	*>10.34.1.0/24	0.0.0.0	*>10.34.1.0/24	0.0.0.0

CONFIGURAÇÃO PARTE 4 - CONFIGURAR IBGP ENTRE R8 E R10 E R10 E R9



```
R10#sh ip route bgp
(NAO APARECE NENHUMA ROTA!!!)
```

```
R10#sh ip bgp
BGP table version is 1, local router ID is 8.9.10.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i 6.0.0.0/24	68.0.0.2	0	100	0	600 i
* i 7.0.0.0/24	78.0.0.2	0	100	0	700 i

1

```
R8
(1) conf t
(2) router bgp 819
(3) neighbor 10.81.0.2 remote-as 819
(4) end
(5) wr
```

2

```
R10
(1) conf t
(2) router bgp 819
(3) neighbor 10.81.0.1 remote-as 819
(4) neighbor 10.91.0.1 remote-as 819
(5) wr
```

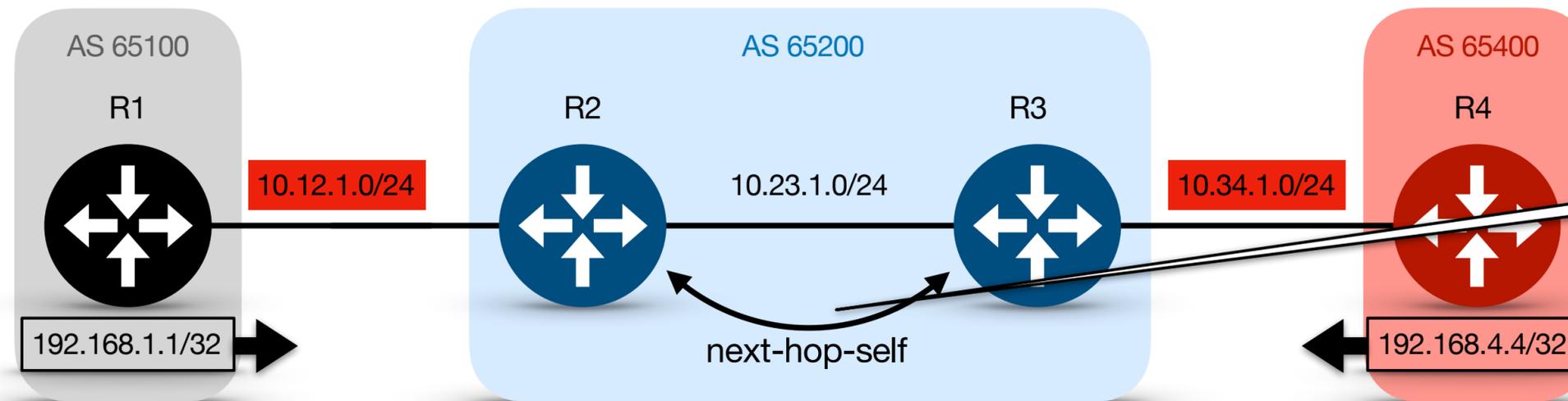
3

```
R9
(1) conf t
(2) router bgp 819
(3) neighbor 10.91.0.2 remote-as 819
(4) end
(5) wr
```

```
R10#sh ip bgp 6.0.0.0/24
BGP routing table entry for 6.0.0.0/24, version 0
Paths: (1 available, no best path)
Not advertised to any peer
Refresh Epoch 1
600
68.0.0.2 (inaccessible) from 10.91.0.1 (68.0.0.1)
Origin IGP, metric 0, localpref 100, valid, internal
rx pathid: 0, tx pathid: 0
```

```
R10#sh ip bgp 7.0.0.0/24
BGP routing table entry for 7.0.0.0/24, version 0
Paths: (1 available, no best path)
Not advertised to any peer
Refresh Epoch 1
700
78.0.0.2 (inaccessible) from 10.81.0.1 (78.0.0.1)
Origin IGP, metric 0, localpref 100, valid, internal
rx pathid: 0, tx pathid: 0
```

NEXT-HOP MANIPULATION



Os prefixos que o R2 receber do eBGP terão o next-hop alterado para o iBGP. (iBGP por iBGP não altera por padrão).

Imagina uma operadora, se ela tiver 500 links desse tipo, ela terá que divulgar 500 redes no seu backbone. Existe uma solução mais elegante!

Outra técnica para garantir que a verificação do endereço do próximo salto passe sem anunciar redes de peering em um protocolo de roteamento envolve a modificação do endereço do próximo salto.

O recurso `nexthop-self` modifica o endereço do próximo salto no NLRI para prefixos BGP externos no endereço IP que origina a sessão BGP.

O comando `neighbor ip-address next-hop-self [all]` é usado para cada vizinho sob a configuração `address family`. O recurso `next-hop-self` não modifica o endereço do próximo salto para prefixos iBGP por padrão. Os nodes IOS podem anexar a palavra-chave opcional `all`, que também modifica o endereço do próximo salto nos prefixos iBGP.

```
R2 (Default IPv4 Address-Family Enabled)
router bgp 65200
neighbor 10.12.1.1 remote-as 65100
neighbor 10.23.1.3 remote-as 65200
neighbor 10.23.1.3 next-hop-self
```

```
R3 (Default IPv4 Address-Family Disabled)
router bgp 65200
no bgp default ipv4-unicast
neighbor 10.23.1.2 remote-as 65200
neighbor 10.34.1.4 remote-as 65400
!
address-family ipv4
neighbor 10.23.1.2 activate
neighbor 10.23.1.2 next-hop-self
neighbor 10.34.1.4 activate
```

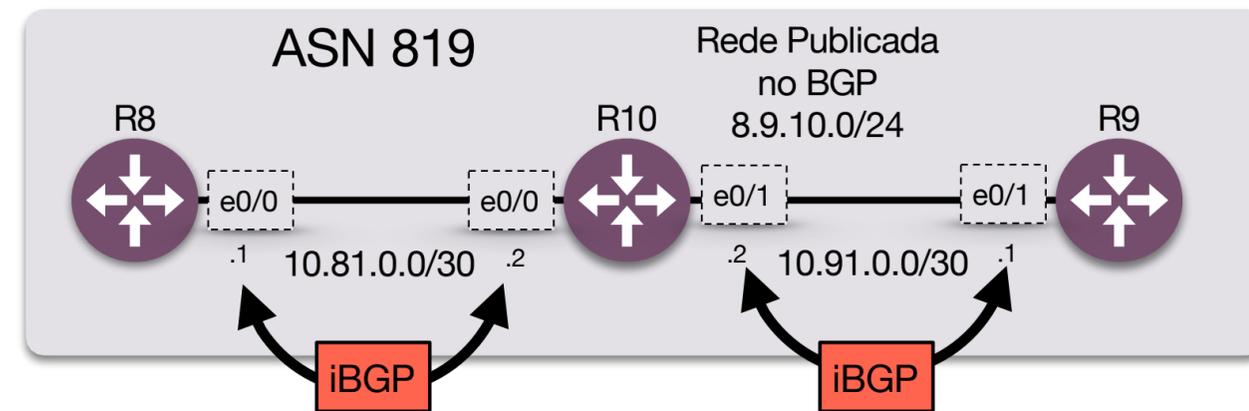
R1	
Network	Next-Hop
*>192.168.1.1/32	0.0.0.0
*>192.168.4.4/32	10.12.1.2

R2	
Network	Next-Hop
*>192.168.1.1/32	10.12.1.1
*>192.168.4.4/32	10.23.1.3

R3	
Network	Next-Hop
*>192.168.1.1/32	10.23.1.2
*>192.168.4.4/32	10.34.1.4

R4	
Network	Next-Hop
*>192.168.1.1/32	10.34.1.3
*>192.168.4.4/32	0.0.0.0

CONFIGURAÇÃO **PARTE 5** - RESOLVER O PROBLEMA DO NEXT-HOP



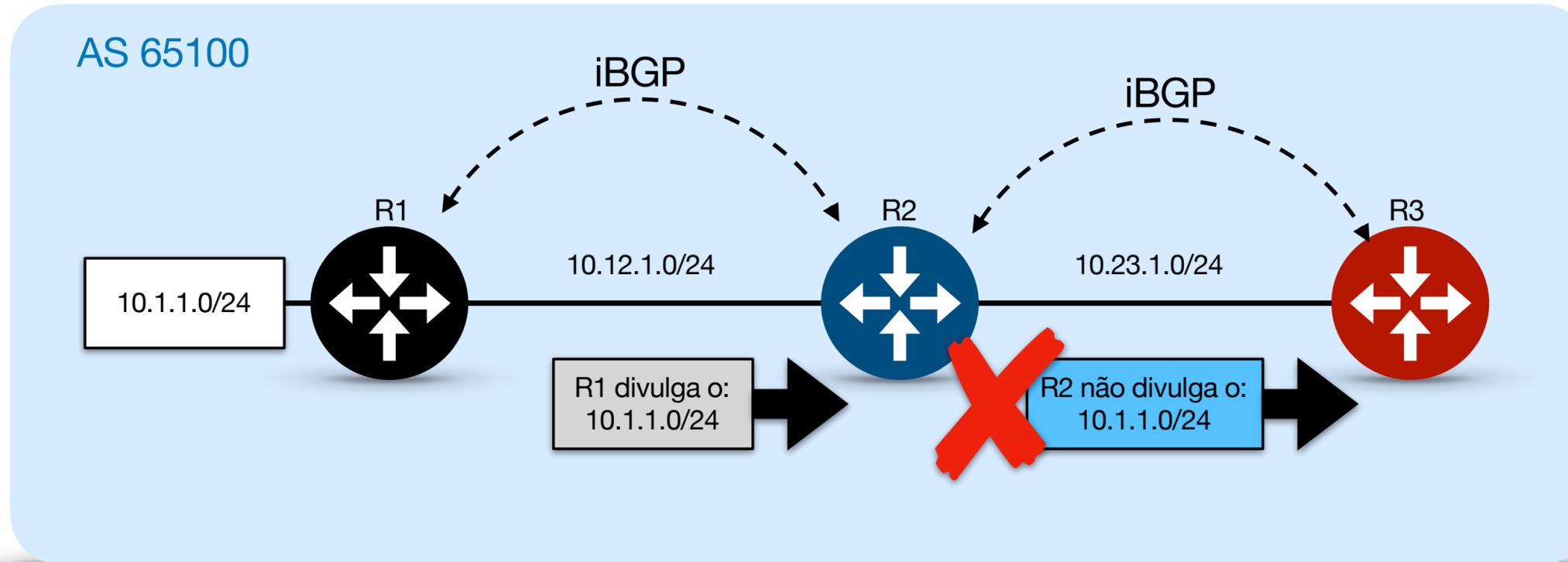
1

```
R8
(1) conf t
(2) router bgp 819
(3) neighbor 10.81.0.2 next-hop-self
(4) end
(5) wr
```

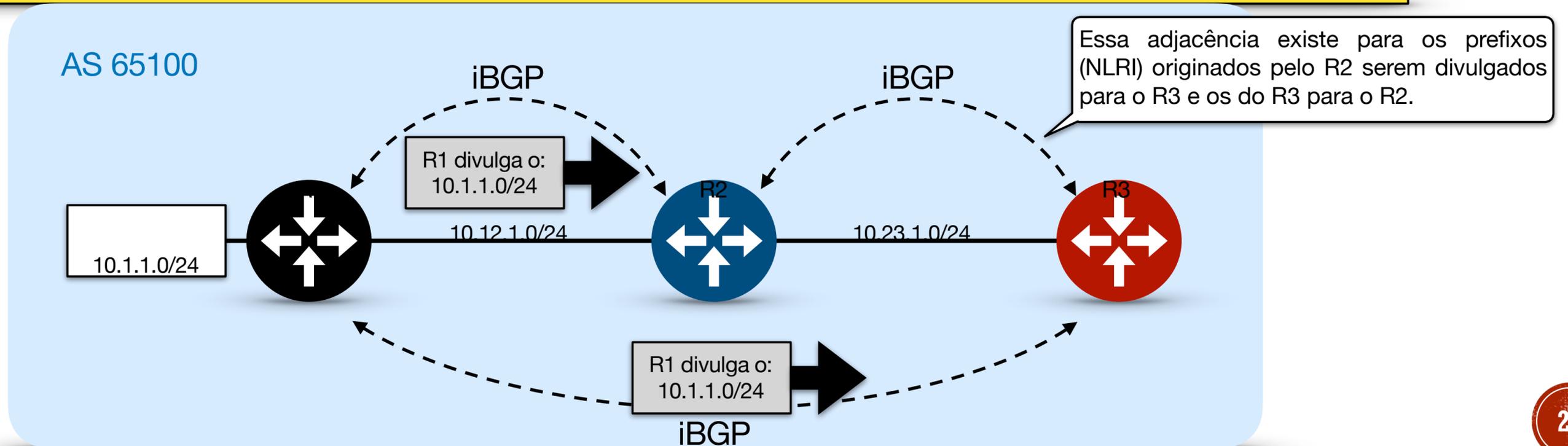
2

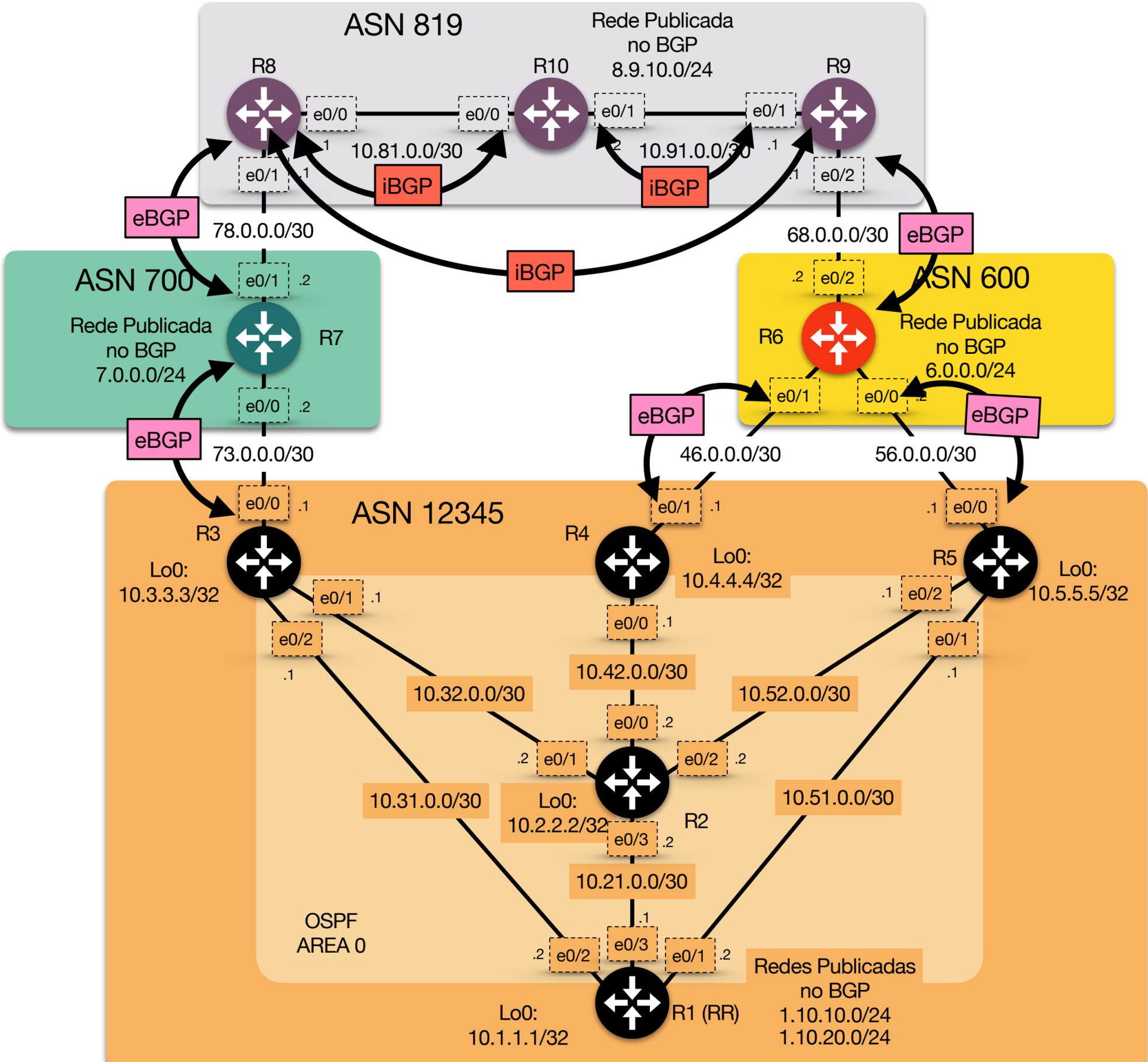
```
R9
(1) conf t
(2) router bgp 819
(3) neighbor 10.91.0.2 next-hop-self
(4) end
(5) wr
```

IBGP FULL MESH REQUIREMENT



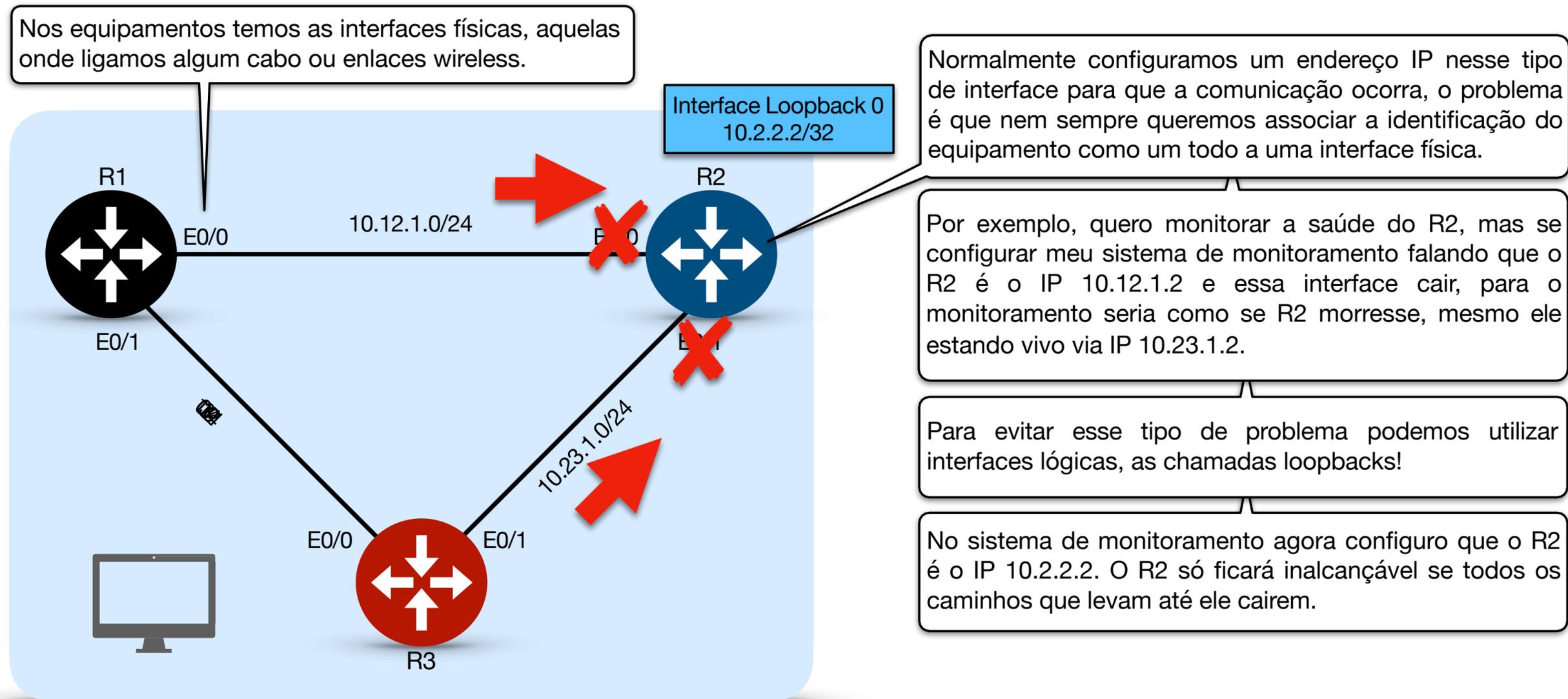
AS_Path pode ser usado para detectar loops para vizinhos eBGP porque o ASN é prefixado ao anunciar para outro AS, só que no iBGP todos usam o mesmo AS (por isso é iBGP né), nenhum outro método existe para detectar loops com sessões iBGP, e a RFC 4271 proíbe o anúncio de um NLRI recebido de um peer iBGP para outro peer iBGP. Para resolver esse problema temos que criar adjacências FULL MESH iBGP, exemplo:





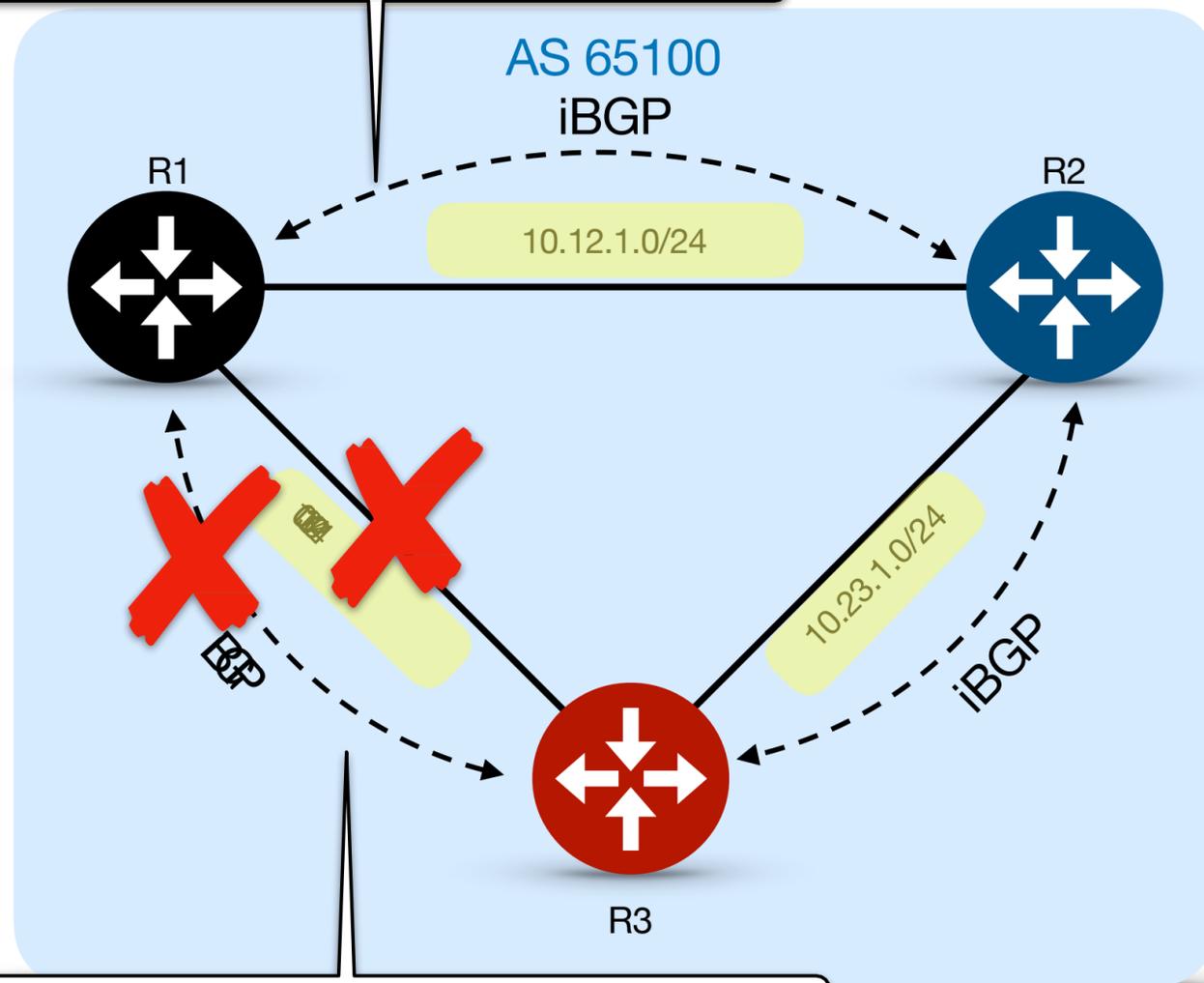
O QUE APRENDEMOS

INTERFACE LOOPBACK: POR QUE USAR?

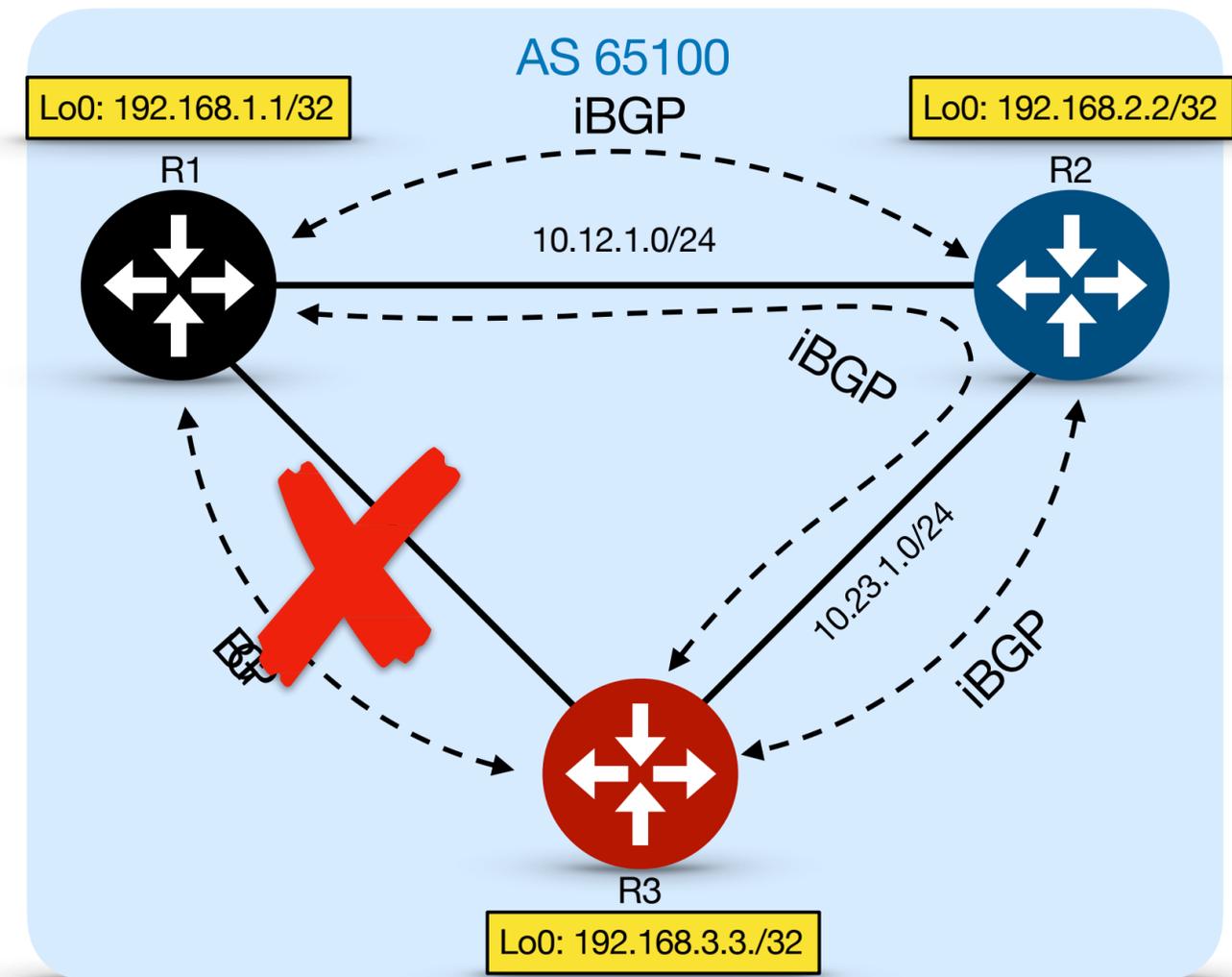


PEERING USANDO LOOPBACK ADDRESSES (IBGP)

Tudo ok, foi feito um FULL MESH iBGP. As sessões BGP são originadas pelo endereço IP primário da interface de saída em direção ao peer BGP por padrão



Mas se esse enlace por exemplo cair, o peer BGP entre R3 e R1 será quebrado, pois a adjacência foi configurada usando o IP das interfaces, para resolver esse problema podemos usar Interfaces Loopback.



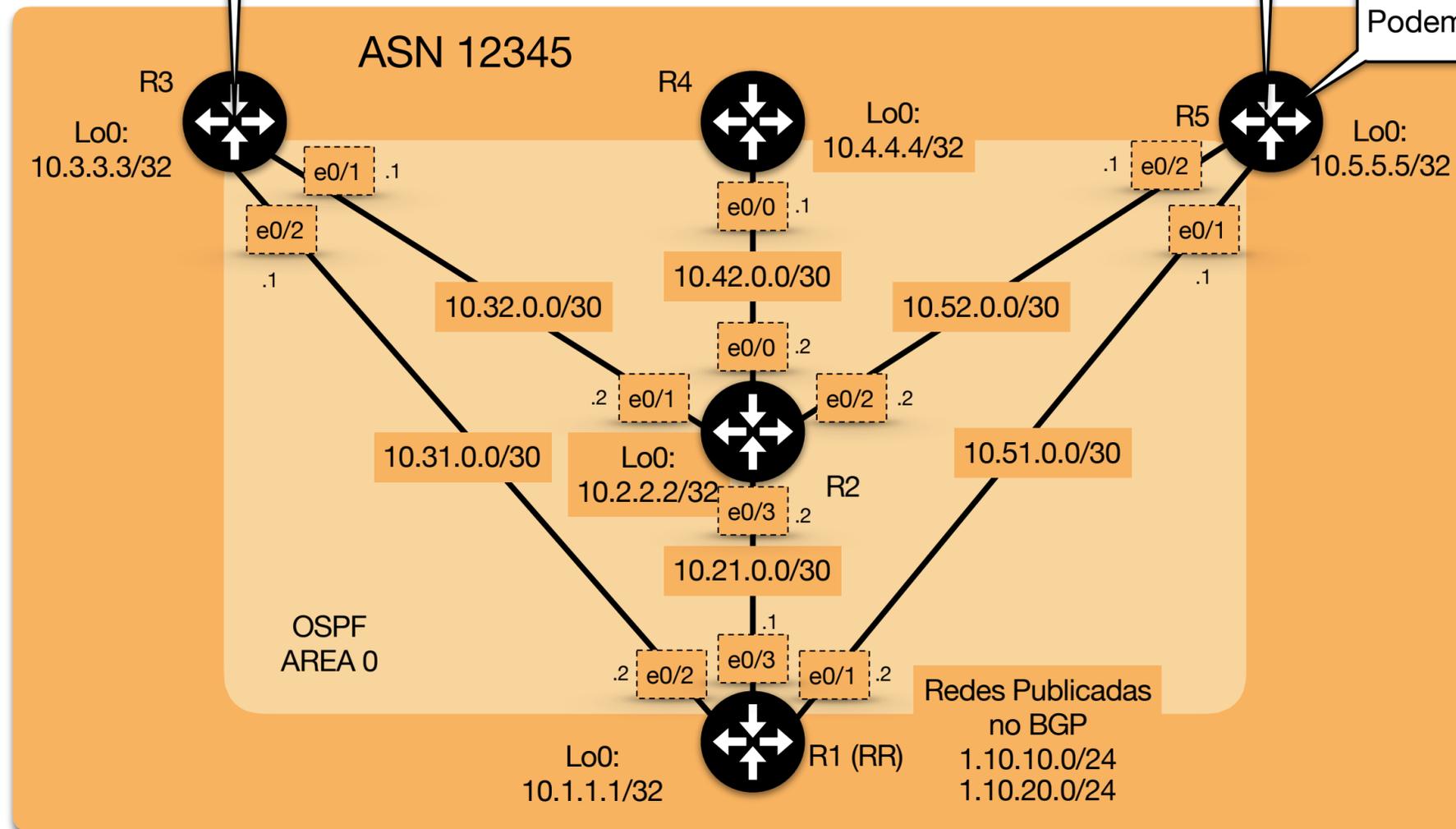
O fornecimento de sessões BGP com interfaces de loopback elimina a necessidade de recalcular o algoritmo best-path do BGP se um link falhar, conforme mostrado anteriormente. Ele também fornece balanceamento de carga automático se houver vários caminhos de custo igual através do IGP para o endereço de loopback. Recurso utilizado normalmente com iBGP.

OSPF NA REDE INTERNA

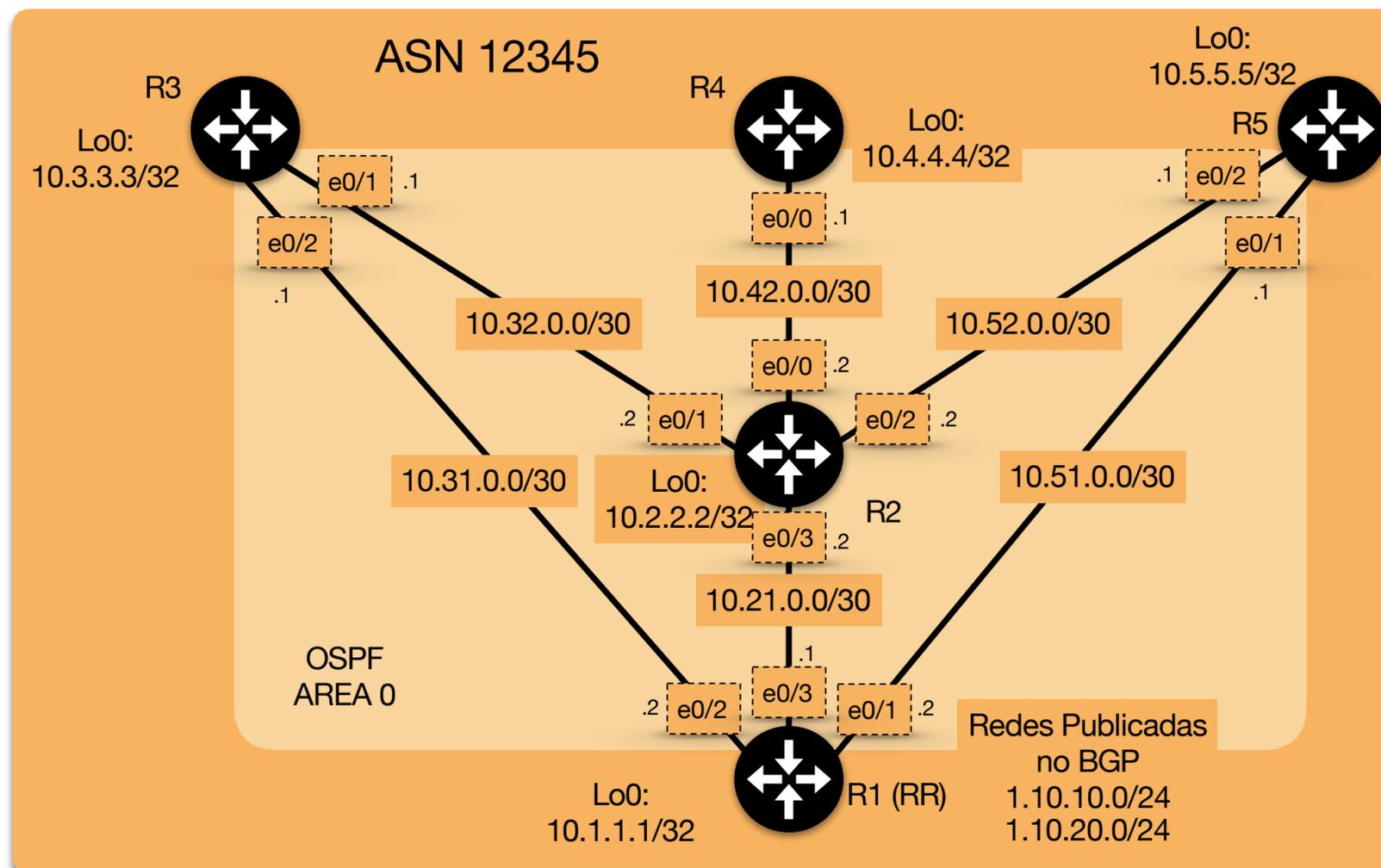
Ok, já vimos que é legal formar o peer iBGP utilizando a loopback quando temos múltiplos caminhos para alcançar os outros dispositivos.

Mas como o R5 conseguirá alcançar a loopback dos demais roteadores? Rota estática nesse cenário seria muito trabalhoso.

Podemos usar o OSPF para fazer esse serviço!



CONFIGURAÇÃO **PARTE 7** - CONFIGURANDO O OSPF PARA DIVULGAR OS IPS DAS LOOPBACKS



R3#

- (1) conf t
- (2) router ospf 10
- (3) network 10.3.3.3 0.0.0.0 area 0
- (4) network 10.31.0.1 0.0.0.0 area 0
- (5) network 10.32.0.1 0.0.0.0 area 0
- (6) exit
- (7) int range e0/1-2
- (8) ip ospf network point-to-point
- (9) end
- (10) wr

R5#

- (1) conf t
- (2) router ospf 10
- (3) network 10.5.5.5 0.0.0.0 area 0
- (4) network 10.52.0.1 0.0.0.0 area 0
- (5) network 10.51.0.1 0.0.0.0 area 0
- (6) exit
- (7) int range e0/1-2
- (8) ip ospf network point-to-point
- (9) end
- (10) wr

R4#

- (1) conf t
- (2) router ospf 10
- (3) network 10.4.4.4 0.0.0.0 area 0
- (4) network 10.42.0.1 0.0.0.0 area 0
- (5) exit
- (6) interface e0/0
- (7) ip ospf network point-to-point
- (8) end
- (9) wr

R2#

- (1) conf t
- (2) router ospf 10
- (3) network 10.2.2.2 0.0.0.0 area 0
- (4) network 10.32.0.2 0.0.0.0 area 0
- (5) network 10.42.0.2 0.0.0.0 area 0
- (6) network 10.52.0.2 0.0.0.0 area 0
- (7) network 10.21.0.2 0.0.0.0 area 0
- (8) exit
- (9) interface range e0/0-3
- (10) ip ospf network point-to-point
- (11) end
- (12) wr

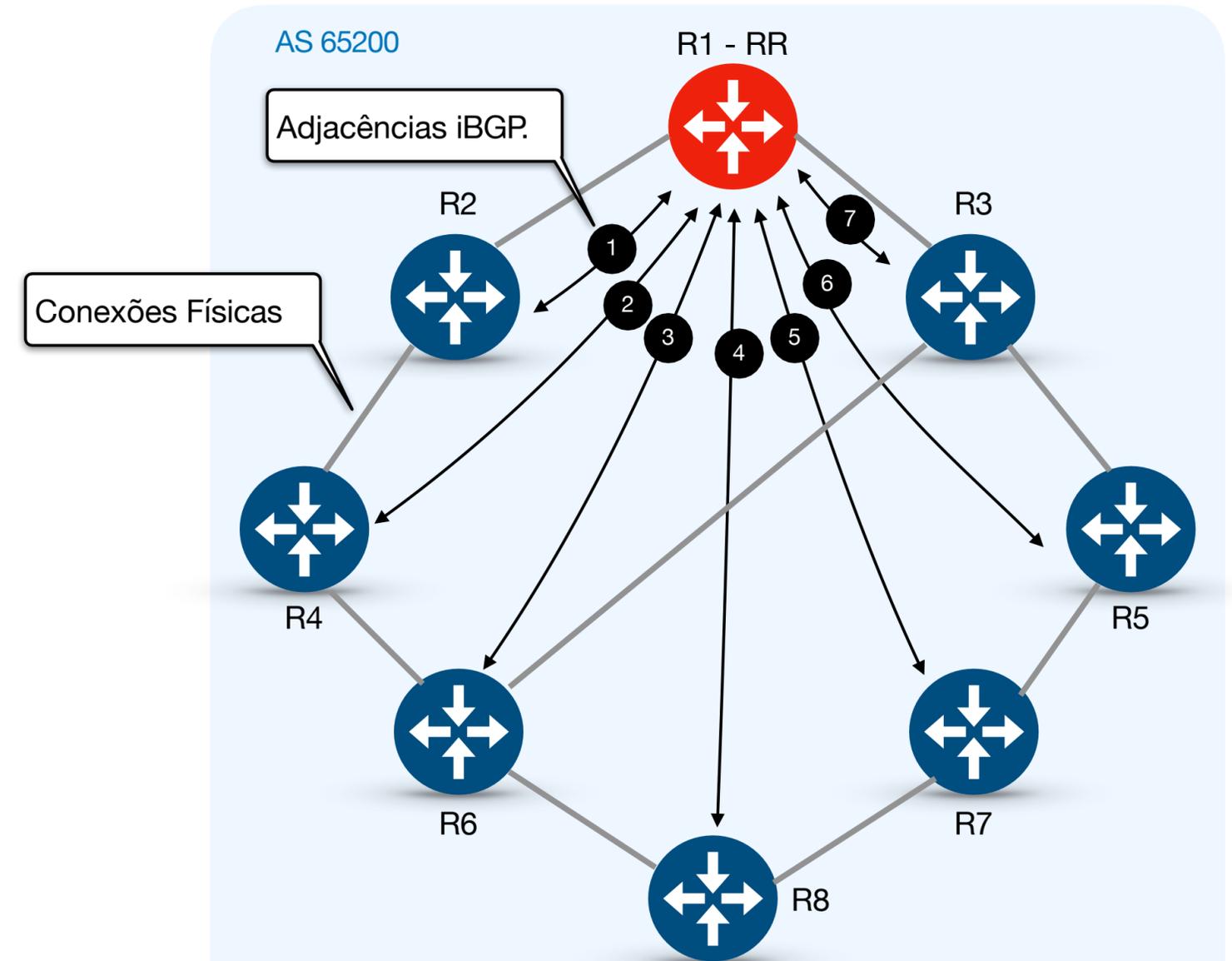
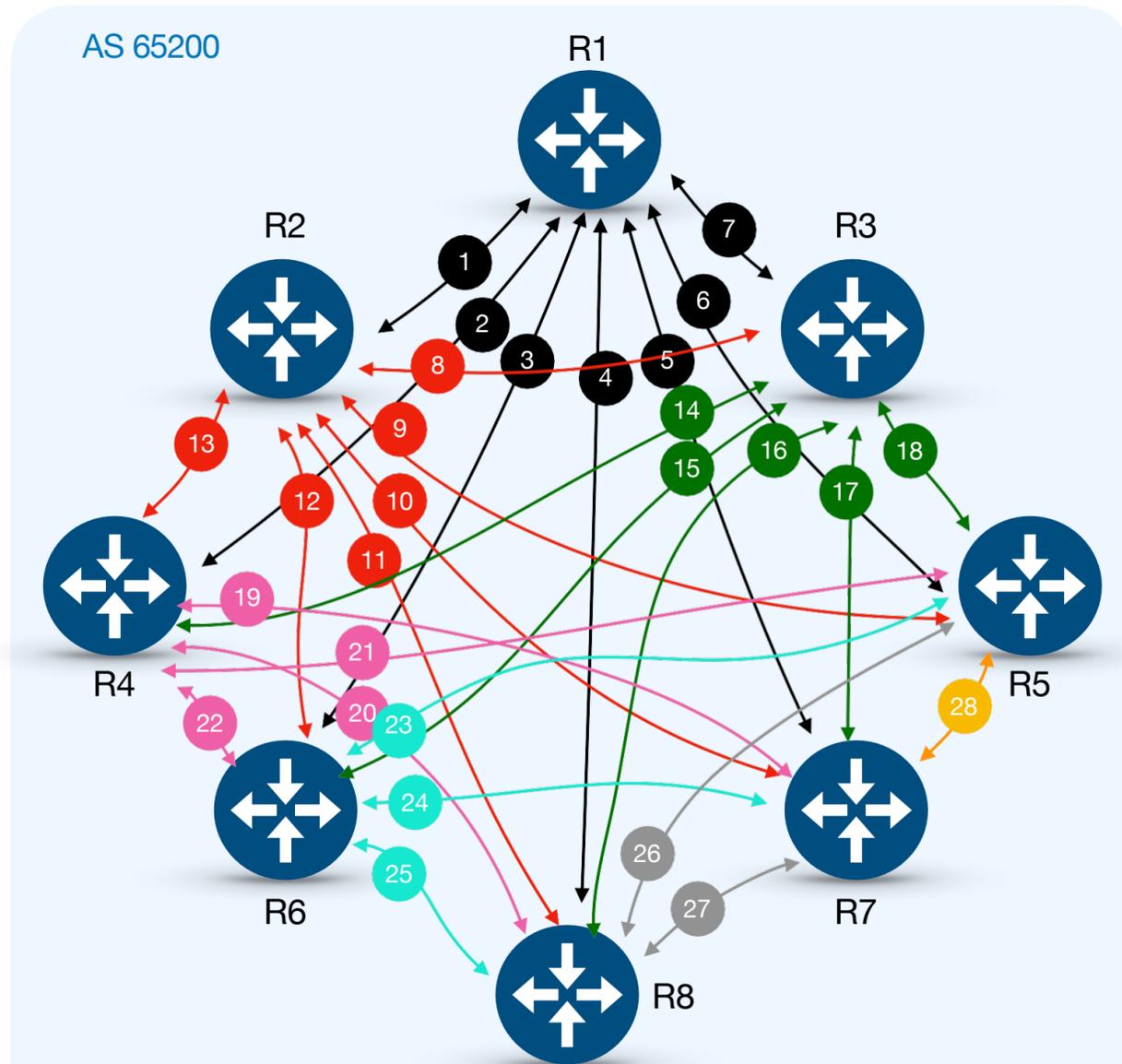
R1#

- (1) conf t
- (2) router ospf 10
- (3) network 10.1.1.1 0.0.0.0 are
- (4) network 10.21.0.1 0.0.0.0 area 0
- (5) network 10.51.0.2 0.0.0.0 area 0
- (6) network 10.31.0.2 0.0.0.0 area 0
- (7) exit
- (8) interface range e0/1-3
- (9) ip ospf network point-to-point
- (10) end
- (11) wr

R2#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.1.1.1	0	FULL/ -	00:00:31	10.21.0.1	Ethernet0/3
10.5.5.5	0	FULL/ -	00:00:39	10.52.0.1	Ethernet0/2
10.4.4.4	0	FULL/ -	00:00:37	10.42.0.1	Ethernet0/0
10.3.3.3	0	FULL/ -	00:00:35	10.32.0.1	Ethernet0/1

MELHORIAS DE ESCALABILIDADE IBGP - ROUTE REFLECTOR



A incapacidade do BGP de anunciar um prefixo aprendido de um peer iBGP para outro peer iBGP pode levar a problemas de escalabilidade em um AS.

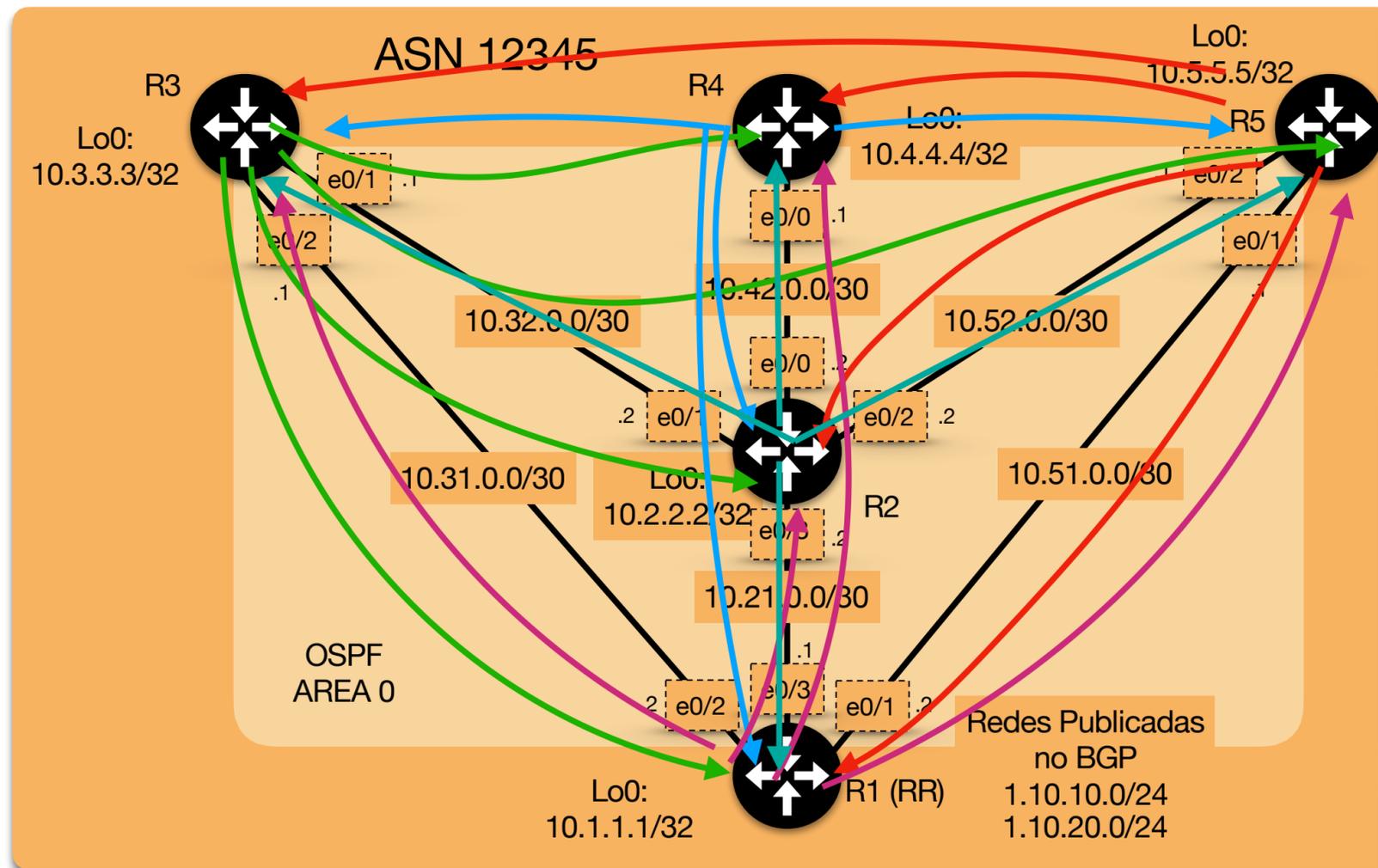
A fórmula $n(n-1)/2$ fornece o número de sessões necessárias, onde n representa o número de roteadores.

Uma topologia full-mesh de 5 roteadores requer 10 sessões iBGP e uma topologia de 8 roteadores requer 28. A escalabilidade do iBGP torna-se um problema para grandes redes.

A RFC 1966 introduz a ideia de que o peering iBGP pode ser configurado de modo que reflita rotas para outro peer iBGP. O roteador que está refletindo as rotas é conhecido como um Router Reflector (RR), e o roteador que está recebendo as rotas refletidas é um route reflector client.

É importante ter em mente que o RR faz apenas o Control Plane, o Data Plane (tráfego) não necessariamente irá passar pelo RR e sim pelo melhor caminho escolhido pelo BGP.

ENTENDENDO O PROBLEMA QUE O ROUTE REFLECTOR RESOLVE NO NOSSO CENÁRIO



E se eu pudesse eleger um ponto focal nessa rede? Ai que entra o router reflector:

```
R3#
(1) router bgp 12345
(2) neighbor 10.4.4.4 remote-as 12345
(3) neighbor 10.4.4.4 next-hop-self
(4) neighbor 10.5.5.5 remote-as 12345
(5) neighbor 10.5.5.5 next-hop-self
(6) neighbor 10.2.2.2 remote-as 12345
(7) neighbor 10.2.2.2 next-hop-self
(8) neighbor 10.1.1.1 remote-as 12345
(9) neighbor 10.1.1.1 next-hop-self
```

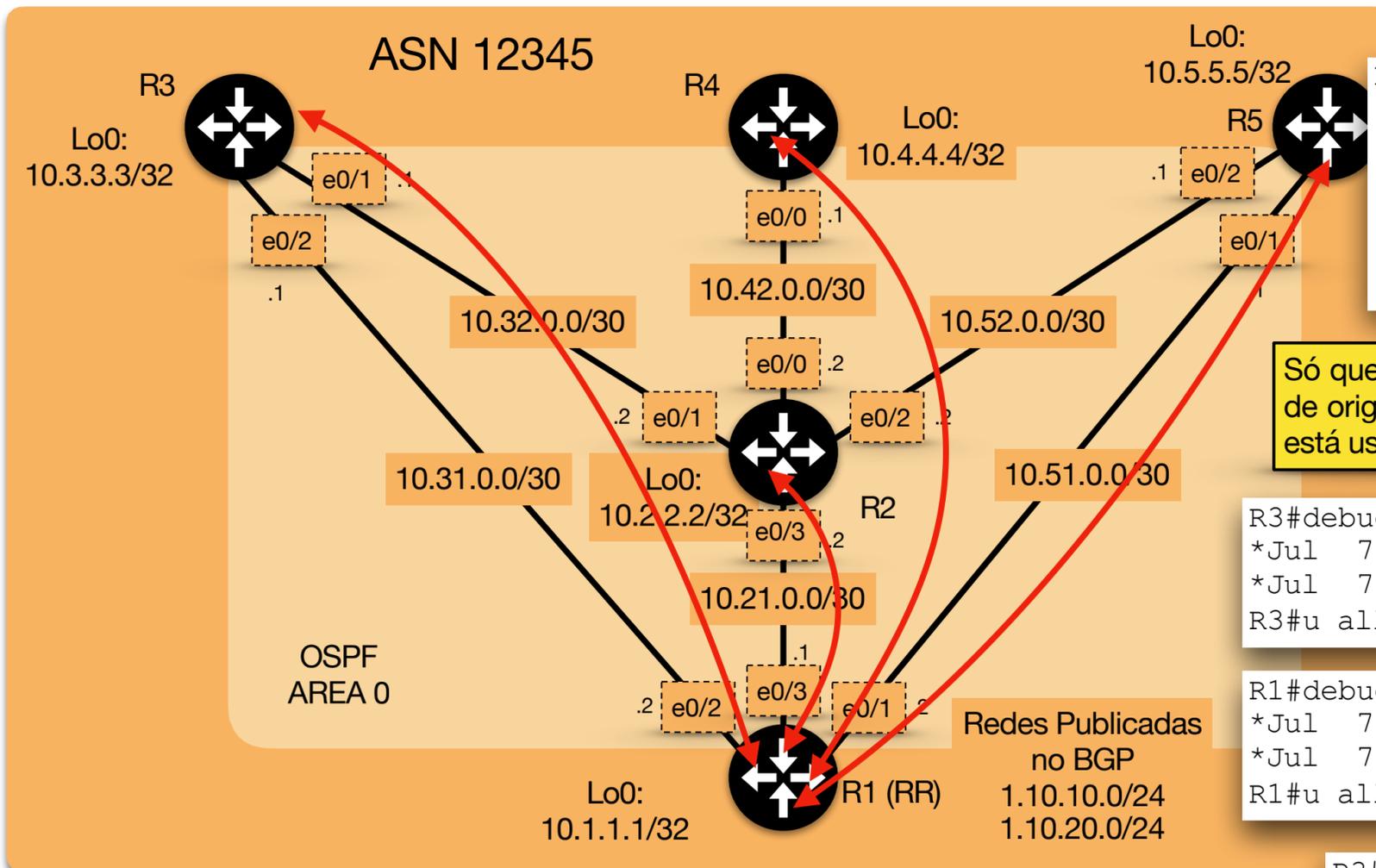
```
R4#
(1) router bgp 12345
(2) neighbor 10.2.2.2 remote-as 12345
(3) neighbor 10.2.2.2 next-hop-self
(4) neighbor 10.5.5.5 remote-as 12345
(5) neighbor 10.5.5.5 next-hop-self
(6) neighbor 10.3.3.3 remote-as 12345
(7) neighbor 10.3.3.3 next-hop-self
(8) neighbor 10.1.1.1 remote-as 12345
(9) neighbor 10.1.1.1 next-hop-self
```

```
R5#
(1) router bgp 12345
(2) neighbor 10.2.2.2 remote-as 12345
(3) neighbor 10.2.2.2 next-hop-self
(4) neighbor 10.4.4.4 remote-as 12345
(5) neighbor 10.4.4.4 next-hop-self
(6) neighbor 10.3.3.3 remote-as 12345
(7) neighbor 10.3.3.3 next-hop-self
(8) neighbor 10.1.1.1 remote-as 12345
(9) neighbor 10.1.1.1 next-hop-self
```

```
R1#
(1) router bgp 12345
(2) neighbor 10.2.2.2 remote-as 12345
(3) neighbor 10.2.2.2 next-hop-self
(4) neighbor 10.4.4.4 remote-as 12345
(5) neighbor 10.4.4.4 next-hop-self
(6) neighbor 10.3.3.3 remote-as 12345
(7) neighbor 10.3.3.3 next-hop-self
(8) neighbor 10.5.5.5 remote-as 12345
(9) neighbor 10.5.5.5 next-hop-self
```

```
R2#
(1) router bgp 12345
(2) neighbor 10.1.1.1 remote-as 12345
(3) neighbor 10.1.1.1 next-hop-self
(4) neighbor 10.4.4.4 remote-as 12345
(5) neighbor 10.4.4.4 next-hop-self
(6) neighbor 10.3.3.3 remote-as 12345
(7) neighbor 10.3.3.3 next-hop-self
(8) neighbor 10.5.5.5 remote-as 12345
(9) neighbor 10.5.5.5 next-hop-self
```

CONFIGURAÇÃO PARTE 8 - ROUTE REFLECTOR E UPDATE SOURCE



1

```
R3#
(1) conf t
(2) router bgp 12345
(3) neighbor 10.1.1.1 remote-as 12345
(4) neighbor 10.1.1.1 next-hop-self
(5) end
(6) wr
```

2

```
R1#
(1) conf t
(2) router bgp 12345
(3) neighbor 10.3.3.3 remote-as 12345
(4) neighbor 10.3.3.3 route-reflector-client
(5) end
(6) wr
```

Só que temos um problema, repare que o peer BGP não irá se formar. A sessão BGP deve ser feita usando o IP de origem da loopback, como não especificamos o IP de origem de forma implícita na configuração do BGP, ele está usando o IP da interface física, e esse IP não é igual ao que colocamos no comando neighbor do BGP.

```
R3#debug ip bgp
*Jul 7 00:43:33.742: BGP: 10.1.1.1 open active, local address 10.31.0.1
*Jul 7 00:43:33.743: BGP: 10.1.1.1 open failed: Connection refused by remote host
R3#u all
```

```
R1#debug ip bgp
*Jul 7 00:42:25.296: BGP: 10.3.3.3 open active, local address 10.31.0.2
*Jul 7 00:42:25.297: BGP: 10.3.3.3 open failed: Connection refused by remote host
R1#u all
```

Corrigindo o problema:

3

```
R3#
(1) conf t
(2) router bgp 12345
(3) neighbor 10.1.1.1 update-source lo0
(4) end
(5) wr
```

4

```
R1#
(1) conf t
(2) router bgp 12345
(3) neighbor 10.3.3.3 update-source lo0
(4) end
(5) wr
```

Configurando os outros clientes RR, nos clientes a configuração não muda nada com relação ao BGP convencional:

```
R4#
(1) conf t
(2) router bgp 12345
(3) neighbor 10.1.1.1 remote-as 12345
(4) neighbor 10.1.1.1 next-hop-self
(5) neighbor 10.1.1.1 update-source lo0
(6) end
(7) wr
```

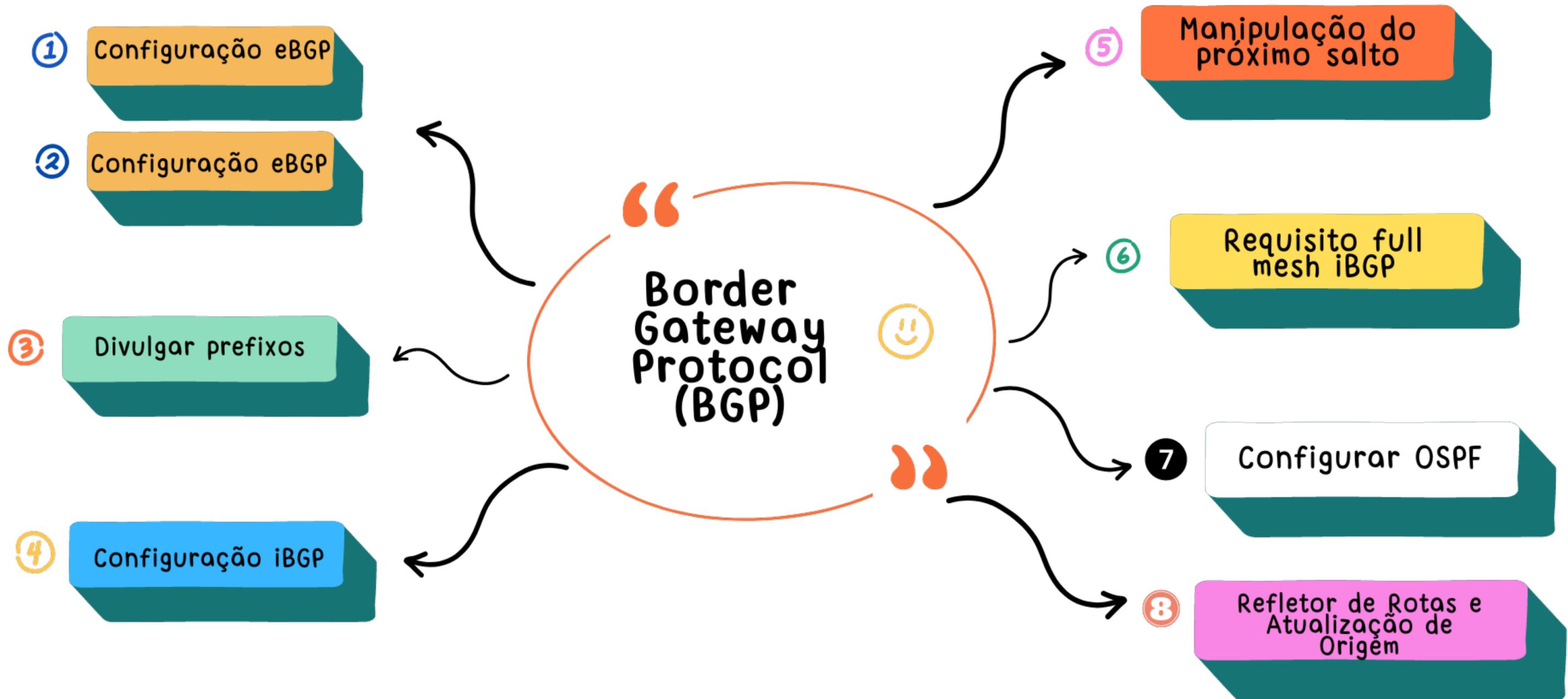
```
R5#
(1) conf t
(2) router bgp 12345
(3) neighbor 10.1.1.1 remote-as 12345
(4) neighbor 10.1.1.1 next-hop-self
(5) neighbor 10.1.1.1 update-source lo0
(6) end
(7) wr
```

```
R2#
(1) conf t
(2) router bgp 12345
(3) neighbor 10.1.1.1 remote-as 12345
(4) neighbor 10.1.1.1 next-hop-self
(5) neighbor 10.1.1.1 update-source lo0
(6) end
(7) wr
```

O QUE FALTA FAZER?

- Fechar adjacências entre:
 - R2 -> R1 (RR).
 - R4 -> R1 (RR).
 - R5 -> R1 (RR).
- Vocês irão perceber que todos roteadores do ASN 12345 vão aprender as rotas 6.0.0.0/24 e 7.0.0.0/24 que foram divulgadas no BGP.
- Podemos ainda divulgar a rede que está por trás do ASN 819 (rede 8.9.10.0/24) e as redes do ASN 12345 (1.10.10.0/24 e 1.10.20.0/24).
- É assim que as redes vão sendo aprendidas na Internet e os roteadores vão conhecendo e instalando em sua Tabela de Rotas as redes que precisam ser alcançadas para que seja possível a comunicação entre os computadores de diferentes Sistemas Autônomos.

CONSIDERAÇÕES FINAIS



DÚVIDAS?

QUESTIONAMENTOS?



COMENTÁRIOS?

AGRADECIMENTOS

- Aos participantes e a comissão organizadora da XIII Secitec do IFG–Câmpus Formosa.
- Ao professor Gustavo Kalau pela disponibilização do material usado nesta oficina sobre o BGP.

OBRIGADO PELA PARTICIPAÇÃO!

- Prof. Mario Lemes (IFG – Campus Formosa)
- Contato: mario.lemes@ifg.edu.br
- Youtube: http://www.youtube.com/c/MarioTeixeiraLemes?sub_confirmation=1
- Currículo lattes: <http://lattes.cnpq.br/4918126641251231>
- Site pessoal: <https://mariotlemes.github.io>
- Slides: <https://encurtador.com.br/gyJP7>



ATÉ A PRÓXIMA! =)