

# Use o Multi-Exit Discriminator para selecionar o melhor caminho para o roteador BGP

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[O Atributo MED](#)

[Exemplo](#)

[O comando bgp deterministic-med](#)

[Examples](#)

[Um roteador BGP com bgp deterministic-med desativado](#)

[Um roteador BGP com bgp deterministic-med habilitado](#)

[Informações Relacionadas](#)

## Introduction

Este documento descreve o `bgp deterministic-med` e explica como ele afeta a seleção de caminho com base no discriminador de várias saídas (MED).

## Prerequisites

## Requirements

Não existem requisitos específicos para este documento.

## Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

## Conventions

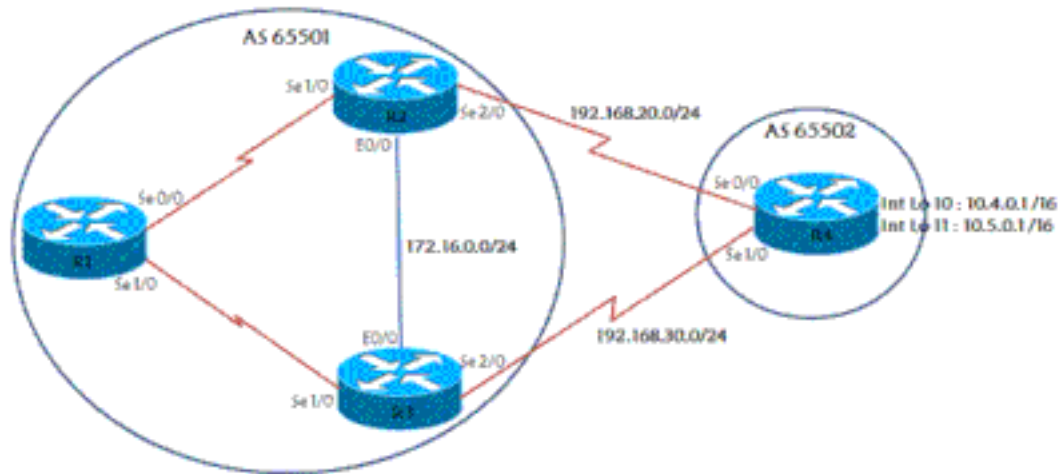
Para obter mais informações sobre convenções de documento, consulte as Convenções de dicas técnicas Cisco.

## O Atributo MED

**MED** é um atributo não transitivo opcional. MED é uma dica para vizinhos externos sobre o caminho preferido em um sistema autônomo (AS) que tem vários pontos de entrada. O MED também é conhecido como métrica externa de uma rota. Um valor MED mais baixo tem preferência sobre um valor mais alto.

Esta seção descreve um exemplo de como usar o MED para influenciar a decisão de roteamento tomada por um AS vizinho.

## Topologia de rede



Topologia de rede

## Exemplo

Neste cenário, AS 65502 é um usuário do ISP que tem AS 65501. O R4 está conectado a dois roteadores diferentes no lado do ISP para fins de redundância e anuncia duas redes ao ISP—10.4.0.0/16 e 10.5.0.0/16. Algumas das configurações relevantes são mostradas nesta seção.

### R4

```
!  
version 12.3  
!  
hostname r4  
!  
ip cef  
!  
!  
interface Loopback10  
 ip address 10.4.0.1 255.255.0.0  
!  
interface Loopback11  
 ip address 10.5.0.1 255.255.0.0  
!  
interface Serial10/0  
 ip address 192.168.20.4 255.255.255.0  
!  
interface Serial11/0  
 ip address 192.168.30.4 255.255.255.0  
!  
router bgp 65502  
 no synchronization  
 bgp log-neighbor-changes  
 network 10.4.0.0 mask 255.255.0.0
```

```
network 10.5.0.0 mask 255.255.0.0
neighbor 192.168.20.2 remote-as 65501
neighbor 192.168.30.3 remote-as 65501
no auto-summary
!
ip classless
!
!
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  exec-timeout 0 0
  login
!
!
end
```

## **R2**

```
!
version 12.3
!
hostname r2
!
ip cef
!
!
interface Loopback0
  ip address 10.2.2.2 255.255.255.255
!
interface Ethernet0/0
  ip address 172.16.0.2 255.255.255.0
!
interface Serial1/0
  ip address 192.168.1.2 255.255.255.0
  serial restart-delay 0
!
interface Serial2/0
  ip address 192.168.20.2 255.255.255.0
  serial restart-delay 0
!
router ospf 1
  log-adjacency-changes
  redistribute connected
  passive-interface Serial2/0
  network 10.2.2.2 0.0.0.0 area 0
  network 172.16.0.2 0.0.0.0 area 0
  network 192.168.1.2 0.0.0.0 area 0
  network 192.168.20.2 0.0.0.0 area 0
!
router bgp 65501
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.1.1.1 remote-as 65501
  neighbor 10.1.1.1 update-source Loopback0
  neighbor 10.3.3.3 remote-as 65501
  neighbor 10.3.3.3 update-source Loopback0
  neighbor 192.168.20.4 remote-as 65502
  no auto-summary
!
ip classless
!
!
line con 0
```

```
exec-timeout 0 0
transport preferred all
transport output all
line aux 0
transport preferred all
transport output all
line vty 0 4
exec-timeout 0 0
login
transport preferred all
transport input all
transport output all
!
end
```

As configurações de R1 e R3 são semelhantes a R2. R3 tem um eBGP que corresponde a R4 e um iBGP que corresponde a R1.

R1 tem um iBGP que corresponde a R2 e a R3. Observe o que as tabelas de BGP de R1, R2 e R3 exibem para as duas redes anunciadas por R4:

```
r2# show ip bgp 10.4.0.1
BGP routing table entry for 10.4.0.0/16, version 7
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.1 10.3.3.3
  65502
    192.168.20.4 from 192.168.20.4 (10.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, external, best
  65502
    192.168.30.4 (metric 74) from 10.3.3.3 (10.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal
```

```
r2# show ip bgp 10.5.0.1
BGP routing table entry for 10.5.0.0/16, version 6
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.1 10.3.3.3
  65502
    192.168.30.4 (metric 74) from 10.3.3.3 (10.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal
  65502
    192.168.20.4 from 192.168.20.4 (10.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

```
r3# show ip bgp 10.4.0.1
BGP routing table entry for 10.4.0.0/16, version 8
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.1 10.2.2.2
  65502
    192.168.20.4 (metric 74) from 10.2.2.2 (10.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal
  65502
    192.168.30.4 from 192.168.30.4 (10.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

```
r3# show ip bgp 10.5.0.1
BGP routing table entry for 10.5.0.0/16, version 10
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
```

```
10.1.1.1 10.2.2.2
65502
  192.168.30.4 from 192.168.30.4 (10.4.4.4)
    Origin IGP, metric 0, localpref 100, valid, external, best
65502
  192.168.20.4 (metric 74) from 10.2.2.2 (10.2.2.2)
    Origin IGP, metric 0, localpref 100, valid, internal
```

```
r1# show ip bgp 10.4.0.1
```

```
BGP routing table entry for 10.4.0.0/16, version 11
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Not advertised to any peer
65502
  192.168.20.4 (metric 128) from 10.2.2.2 (10.2.2.2)
    Origin IGP, metric 0, localpref 100, valid, internal, best
65502
  192.168.30.4 (metric 128) from 10.3.3.3 (10.3.3.3)
    Origin IGP, metric 0, localpref 100, valid, internal
```

```
r1# show ip bgp 10.5.0.1
```

```
BGP routing table entry for 10.5.0.0/16, version 10
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Not advertised to any peer
65502
  192.168.30.4 (metric 128) from 10.3.3.3 (10.3.3.3)
    Origin IGP, metric 0, localpref 100, valid, internal
65502
  192.168.20.4 (metric 128) from 10.2.2.2 (10.2.2.2)
    Origin IGP, metric 0, localpref 100, valid, internal, best
```

R2 e R3 escolhem como melhor caminho a rota externa de R4, o que é esperado com base no algoritmo de seleção de melhor caminho BGP. Consulte o [algoritmo de seleção de melhor caminho BGP para obter mais informações](#).

Da mesma forma, R1 escolhe R2 para acessar as 2 redes, o que está de acordo com a regra do melhor caminho BGP—selecione o caminho com o menor ID de roteador. Como o ID do roteador R2 é 10.2.2.2 e o ID do roteador R3 é 10.3.3.3, R2 é escolhido. Nessa configuração básica, todo o tráfego para as duas redes no AS 65502 passa de R1 a R2 e depois para R4 por padrão. Agora, suponha que R4 queira balancear a carga do tráfego que recebe do AS 65501. Para fazer isso sem nenhuma modificação do ISP R4, você configura o R4 para utilizar o MED para forçar o tráfego de uma rede para um caminho abaixo e o tráfego da outra rede para o outro caminho.

Esta é a configuração de R4 depois que você aplica a configuração necessária:

## R4

```
!
version 12.3
!
hostname r4
!
ip cef
!
!
!
interface Loopback10
 ip address 10.4.0.1 255.255.0.0
!
interface Loopback11
 ip address 10.5.0.1 255.255.0.0
!
interface Serial0/0
```

```

ip address 192.168.20.4 255.255.255.0
!
interface Serial1/0
ip address 192.168.30.4 255.255.255.0
!
router bgp 65502
no synchronization
bgp log-neighbor-changes
network 10.4.0.0 mask 255.255.0.0
network 10.5.0.0 mask 255.255.0.0
neighbor 192.168.20.2 remote-as 65501
neighbor 192.168.20.2 route-map setMED-R2 out
neighbor 192.168.30.3 remote-as 65501
neighbor 192.168.30.3 route-map setMED-R3 out
no auto-summary
!
ip classless
no ip http server
!
!
access-list 1 permit 10.4.0.0 0.0.255.255
access-list 2 permit 10.5.0.0 0.0.255.255
!
route-map setMED-R3 permit 10
match ip address 1
set metric 200
!
route-map setMED-R3 permit 20
match ip address 2
set metric 100

!--- The route-map MED-R3 is applying a MED of 200 to the 10.4.0.0/16
!--- network and a MED of 100 to the 10.5.0.0/16 network.
!--- The route-map is being applied outbound towards R3. ! route-map setMED-R2 permit 10 match ip address 1
set metric 100 ! route-map setMED-R2 permit 20 match ip address 2 set metric 200 !--- The route-map MED-R2 is
applying a MED of 100 to the 10.4.0.0/16
!--- network and a MED of 200 to the 10.5.0.0/16 network.
!--- The route-map is being applied outbound towards R2. !!! line con 0 exec-timeout 0 0 line aux 0 1
vty 0 4 exec-timeout 0 0 login !! end

```

**Observação:** você precisa limpar a sessão BGP com o comando `clear ip bgp * soft out`, por exemplo, para fazer com que essas configurações executem ações.

R1 agora vê a rota em R2 como o melhor caminho para a rede 10.4.0.0/16 porque a atualização recebida de R2 tem um MED de 100 versus um MED de 200, que é o que R3 anuncia. Da mesma forma, R1 usa R3 e o link R3 - R4 para acessar 10.5.0.0/16:

```

r1# show ip bgp 10.4.0.1
BGP routing table entry for 10.4.0.0/16, version 14
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x800
Not advertised to any peer
65502
192.168.20.4 (metric 128) from 10.2.2.2 (10.2.2.2)
Origin IGP, metric 100, localpref 100, valid, internal, best
r1#sh ip bgp 10.5.0.1
BGP routing table entry for 10.5.0.0/16, version 13
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x800
Not advertised to any peer

```

```
65502
 192.168.30.4 (metric 128) from 10.3.3.3 (10.3.3.3)
   Origin IGP, metric 100, localpref 100, valid, internal, best
```

Examine a tela do R2:

```
r2# show ip bgp 10.4.0.1
BGP routing table entry for 10.4.0.0/16, version 10
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.1 10.3.3.3
  65502
    192.168.20.4 from 192.168.20.4 (10.4.4.4)
      Origin IGP, metric 100, localpref 100, valid, external, best
```

```
r2# show ip bgp 10.5.0.1
BGP routing table entry for 10.5.0.0/16, version 11
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    192.168.20.4
  65502
    192.168.30.4 (metric 74) from 10.3.3.3 (10.3.3.3)
      Origin IGP, metric 100, localpref 100, valid, internal, best
  65502
    192.168.20.4 from 192.168.20.4 (10.4.4.4)
      Origin IGP, metric 200, localpref 100, valid, external
```

A razão pela qual R2 mostra apenas um caminho para 10.4.0.0/16 é porque R3 retira (envia uma atualização com métrica inalcançável) a atualização para 10.4.0.0/16 uma vez que percebe que R3 usa R2 para acessar 10.4.0.0/16 (depois que você executa o melhor caminho BGP em todos os caminhos disponíveis):

```
r3# show ip bgp 10.4.0.0
BGP routing table entry for 10.4.0.0/16, version 20
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    192.168.30.4
  65502
    192.168.20.4 (metric 74) from 10.2.2.2 (10.2.2.2)
      Origin IGP, metric 100, localpref 100, valid, internal, best
  65502
    192.168.30.4 from 192.168.30.4 (10.4.4.4)
      Origin IGP, metric 200, localpref 100, valid, external
```

Isso permite que R2 salve alguma memória, pois não precisa armazenar essas informações inúteis. Caso a sessão BGP entre R2 e R4 falhe, R2 enviaria uma atualização inalcançável para R3 para 10.4.0.0/16. Essa atualização acionaria R3 para enviar uma atualização com a rota de R3 para 10.4.0.0/16 através de R4 para R2. R2 pode começar a rotear via R3.

## O comando `bgp deterministic-med`

Se você habilitar o comando `bgp deterministic-med` remove qualquer dependência temporal de decisões de melhor caminho baseadas em MED. Ele garante que uma comparação MED precisa seja feita em todas as rotas recebidas do mesmo sistema autônomo (AS).

Se você desativar o `bgp deterministic-med`, a ordem na qual as rotas são recebidas pode afetar as decisões de melhor caminho baseadas em MED. Isso pode ocorrer quando a mesma rota é

recebida de vários ASs ou sub-ASs de confederação, com exatamente o mesmo comprimento de caminho, mas MEDs diferentes.

## Examples

Por exemplo, considere as próximas rotas:

```
entry1: ASPATH 1, MED 100, internal, IGP metric to NEXT_HOP 10
entry2: ASPATH 2, MED 150, internal, IGP metric to NEXT_HOP 5
entry3: ASPATH 1, MED 200, external
```

A ordem na qual as rotas BGP foram recebidas é entrada3, entrada2 e entrada1 (entrada3 é a entrada mais antiga na tabela BGP e entrada1 é a mais nova).

### Um roteador BGP com `bgp deterministic-med` desativado

Um roteador BGP com `bgp deterministic-med` desativado escolhe a entrada2 em vez da entrada1, devido a uma métrica IGP mais baixa para acessar o NEXT\_HOP (MED não foi usado nesta decisão porque a entrada1 e a entrada2 são de dois ASs diferentes). Em seguida, ele prefere a entrada3 em vez da entrada2, pois ela é externa. No entanto, a entrada3 tem um MED maior que a entrada1. Para obter mais informações sobre critérios de seleção de caminho BGP, consulte [Algoritmo de seleção de melhor caminho BGP](#).

### Um roteador BGP com `bgp deterministic-med` habilitado

Nesse caso, as rotas do mesmo AS são agrupadas e as melhores entradas de cada grupo são comparadas. No exemplo dado, há dois ASs, AS 1 e AS 2.

```
Group 1:  entry1: ASPATH 1, MED 100, internal, IGP metric to NEXT_HOP 10
          entry3: ASPATH 1, MED 200, external
Group 2:  entry2: ASPATH 2, MED 150, internal, IGP metric to NEXT_HOP 5
```

No Grupo 1, o melhor caminho é a entrada 1 devido ao MED inferior (o MED é usado nesta decisão já que os caminhos são do mesmo AS). No Grupo 2, há apenas uma entrada (entrada2). O melhor caminho então é determinado com uma comparação dos vencedores de cada grupo (MED não é usado nesta comparação por padrão porque os vencedores de cada grupo são de AS diferentes. Quando você habilita `bgp always-compare-med` ele altera esse comportamento padrão). Agora, quando você compara o entry1 (o vencedor do Grupo 1) e o entry2 (o vencedor do Grupo 2), o entry2 pode ser o vencedor, já que tem a melhor métrica IGP para o próximo salto.

Se `bgp always-compare-med` O também foi habilitado quando você compara o item 1 (o vencedor do Grupo 1) e o item 2 (o vencedor do Grupo 2), o item 1 pode ser o vencedor devido ao MED inferior.

A Cisco recomenda que você habilite `bgp always-compare-med` em todas as novas implantações de rede. Além disso, se `bgp always-compare-med` estiver ativado, as decisões MED do BGP são sempre determinísticas.

Para obter mais informações sobre o `bgp deterministic-med` e o `bgp always-compare-med` consulte [Como o comando `bgp deterministic-med` difere do `comando bgp always-compare-med`](#).



## Informações Relacionadas

- [Configuração de exemplo para o BGP com dois provedores de serviço diferentes \(multilocal\)](#)
- [Suporte IP/BGP](#)
- [Estudos de caso de BGP](#)
- [Suporte técnico e downloads da Cisco](#)

## Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.