

# MDT dati e origine dual-homed in mVPN

## Sommario

[Introduzione](#)

[Il problema](#)

[Meccanismo di asserzione sull'MDT di default](#)

[Conclusioni](#)

[Meccanismo di asserzione con MDT dati](#)

[Conclusioni](#)

## Introduzione

Questo documento descrive mVPN (Multicast Virtual Provider Network) con origine dual-homed e Data MDT (Multicast Distribution Tree). Per illustrare il comportamento, viene usato un esempio in Cisco IOS<sup>®</sup>.

## Il problema

Se un'origine nel mondo mVPN è dual-homed per due router Ingress Provider Edge (PE), potrebbe essere possibile per i due router PE in ingresso inoltrare il traffico per uno (S,G) nel cloud Multiprotocol Label Switching (MPLS). Ciò è possibile, ad esempio, se vi sono due router PE in uscita e ciascun router RPF (Reverse Path Forwarding) viene indirizzato a un router PE in entrata diverso. Se entrambi i router PE in ingresso avanzano sull'MDT predefinito, il meccanismo di asserzione si attiva e un PE in ingresso vince il meccanismo di asserzione e l'altro perde in modo che un solo PE in ingresso continui a inoltrare il Cliente (C-) (S,G) sull'MDT. Tuttavia, se per qualsiasi motivo il meccanismo di asserzione non è stato avviato sull'MDT predefinito, è possibile per entrambi i router PE in ingresso iniziare a trasmettere il traffico multicast C-(S,G) su un MDT dati che hanno avviato. Poiché il traffico non si trova più sull'MDT predefinito, ma sugli MDT dati, entrambi i router PE in entrata non ricevono il traffico C-(S,G) l'uno dall'altro sull'interfaccia MDT/Tunnel. Ciò può causare traffico duplicato persistente a valle. Questo documento spiega la soluzione al problema.

## Meccanismo di asserzione sull'MDT di default

Le informazioni di questa sezione sono valide per l'MDT predefinito, indipendentemente dal protocollo della struttura ad albero principale. Il protocollo dell'albero principale scelto è il protocollo PIM (Protocol Independent Multicast).

Negli esempi viene utilizzato Cisco IOS, ma tutto ciò che viene menzionato si applica ugualmente a Cisco IOS-XR. Tutti i gruppi multicast utilizzati sono gruppi multicast specifici dell'origine (SSM).

Vedere la figura 1. Dual-Homed-Source-1. Sono disponibili due router PE in ingresso (PE1 e PE2) e due router PE in uscita (PE3 e PE4). L'origine si trova in CE1 con indirizzo IP 10.100.1.6. CE1 ha una doppia posizione rispetto a PE1 e PE2.

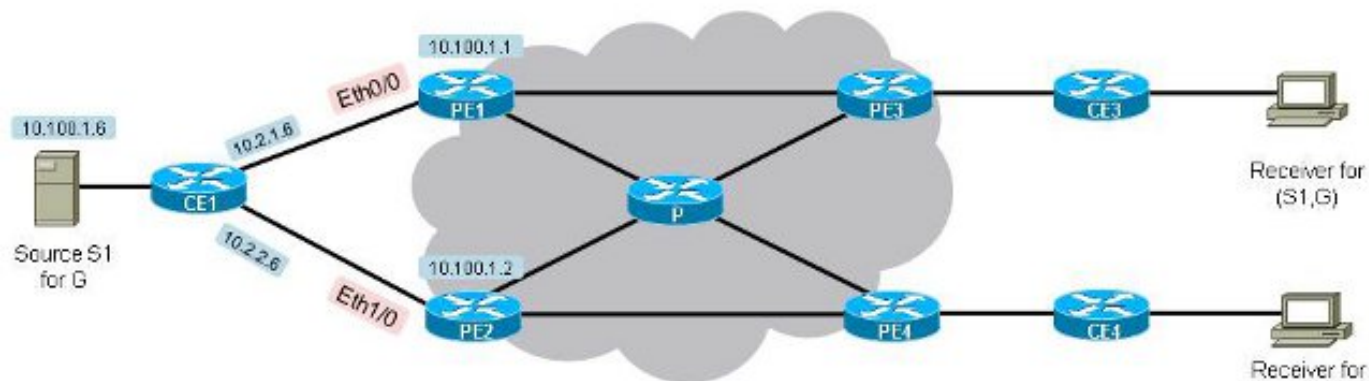


Figura 1. Dual-Homed-Source-1

La configurazione su tutti i router PE (il Distinguotore di route (RD) può essere diverso sui router PE) è:

```
vrf definition one
 rd 1:1
 !
 address-family ipv4
 mdt default 232.10.10.10
 route-target export 1:1
 route-target import 1:1
 exit-address-family
 !
```

Per fare in modo che entrambi i router PE in ingresso inizino ad inoltrare il flusso multicast (10.100.1.6,232.1.1.1) in uscita sull'MDT predefinito, è necessario che entrambi ricevano un join da un PE in uscita. Esaminare la topologia nella Figura 1. Origine dual-home 1. È possibile notare che per impostazione predefinita, se tutti i costi dei collegamenti agli spigoli sono uguali e tutti i costi dei collegamenti centrali sono uguali, PE3 eseguirà RPF verso PE1 e PE4 eseguirà RPF verso PE2 per (10.100.1.6,232.1.1.1). Entrambi fanno riferimento al loro PE più vicino. Questo output conferma quanto segue:

```
PE3#show ip rpf vrf one 10.100.1.6
RPF information for ? (10.100.1.6)
 RPF interface: Tunnel0
 RPF neighbor: ? (10.100.1.1)
 RPF route/mask: 10.100.1.6/32
 RPF type: unicast (bgp 1)
 Doing distance-preferred lookups across tables
 BGP originator: 10.100.1.1
 RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

PE3 dispone di RPF per PE1.

```
PE4#show ip rpf vrf one 10.100.1.6
RPF information for ? (10.100.1.6)
RPF interface: Tunnel0
RPF neighbor: ? (10.100.1.2)
RPF route/mask: 10.100.1.6/32
RPF type: unicast (bgp 1)
Doing distance-preferred lookups across tables
BGP originator: 10.100.1.2
RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

PE4 dispone di RPF per PE2. Il motivo per cui PE3 sceglie PE1 come router adiacente RPF è che la route unicast verso 10.100.1.6/32 in VRF (Virtual Routing/Forwarding) è la migliore tramite PE1. PE3 riceve effettivamente la route 10.100.1.6/32 da PE1 e PE2. Tutti i criteri nell'algoritmo di calcolo del miglior percorso BGP (Border Gateway Protocol) sono uguali, ad eccezione del costo per l'indirizzo dell'hop successivo BGP.

```
PE3#show bgp vpnv4 unicast vrf one 10.100.1.6/32
BGP routing table entry for 1:3:10.100.1.6/32, version 333
Paths: (2 available, best #1, table one)
Advertised to update-groups:
  21
Refresh Epoch 1
Local, imported path from 1:1:10.100.1.6/32 (global)
  10.100.1.1 (metric 11) (via default) from 10.100.1.5 (10.100.1.5)
  Origin incomplete, metric 11, localpref 100, valid, internal,best
  Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
    OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.4.1:0
  Originator: 10.100.1.1, Cluster list: 10.100.1.5
  Connector Attribute: count=1
    type 1 len 12 value 1:1:10.100.1.1
  mpls labels in/out nolabel/32
  rx pathid: 0, tx pathid: 0x0
Refresh Epoch 1
Local, imported path from 1:2:10.100.1.6/32 (global)
  10.100.1.2 (metric 21) (via default) from 10.100.1.5 (10.100.1.5)
  Origin incomplete, metric 11, localpref 100, valid, internal
  Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
    OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.2.2:0
  Originator: 10.100.1.2, Cluster list: 10.100.1.5
  Connector Attribute: count=1
    type 1 len 12 value 1:2:10.100.1.2
  mpls labels in/out nolabel/29
  rx pathid: 0, tx pathid: 0
```

```
PE4#show bgp vpnv4 unicast vrf one 10.100.1.6/32
BGP routing table entry for 1:4:10.100.1.6/32, version 1050
Paths: (2 available, best #2, table one)
Advertised to update-groups:
  2
Refresh Epoch 1
Local, imported path from 1:1:10.100.1.6/32 (global)
  10.100.1.1 (metric 21) (via default) from 10.100.1.5 (10.100.1.5)
  Origin incomplete, metric 11, localpref 100, valid, internal
  Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
    OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.4.1:0
  Originator: 10.100.1.1, Cluster list: 10.100.1.5
  Connector Attribute: count=1
    type 1 len 12 value 1:1:10.100.1.1
  mpls labels in/out nolabel/32
  rx pathid: 0, tx pathid: 0
Refresh Epoch 1
```

```

Local, imported path from 1:2:10.100.1.6/32 (global)
 10.100.1.2 (metric 11) (via default) from 10.100.1.5 (10.100.1.5)
Origin incomplete, metric 11, localpref 100, valid, internal, best
Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
  OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.2.2:0
Originator: 10.100.1.2, Cluster list: 10.100.1.5
Connector Attribute: count=1
  type 1 len 12 value 1:2:10.100.1.2
mpls labels in/out nolabel/29
rx pathid: 0, tx pathid: 0x0

```

Il miglior percorso scelto da PE3 è il percorso annunciato da PE1 perché presenta il costo IGP (Interior Gateway Protocol) più basso (11) rispetto al costo IGP (21) verso PE2. Per PE4 è il contrario. La topologia rivela che da PE3 a PE1 esiste un solo hop, mentre da PE3 a PE2 esistono due hop. Poiché tutti i collegamenti hanno lo stesso costo IGP, PE3 sceglie il percorso di PE1 come migliore.

La base MRIB (Multicast Routing Information Base) per (10.100.1.6,232.1.1.1) ha il seguente aspetto su PE1 e PE2 quando non è ancora presente traffico multicast:

```

PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
  L - Local, P - Pruned, R - RP-bit set, F - Register flag,
  T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
  X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
  U - URD, I - Received Source Specific Host Report,
  Z - Multicast Tunnel, z - MDT-data group sender,
  Y - Joined MDT-data group, y - Sending to MDT-data group,
  G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
  N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
  Q - Received BGP S-A Route, q - Sent BGP S-A Route,
  V - RD & Vector, v - Vector, p - PIM Joins on route,
  x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:00:12/00:03:17, flags: sT
Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
Outgoing interface list:
  Tunnel0, Forward/Sparse, 00:00:12/00:03:17

```

```

PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
  L - Local, P - Pruned, R - RP-bit set, F - Register flag,
  T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
  X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
  U - URD, I - Received Source Specific Host Report,
  Z - Multicast Tunnel, z - MDT-data group sender,
  Y - Joined MDT-data group, y - Sending to MDT-data group,
  G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
  N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
  Q - Received BGP S-A Route, q - Sent BGP S-A Route,
  V - RD & Vector, v - Vector, p - PIM Joins on route,
  x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

```

```
(10.100.1.6, 232.1.1.1), 00:00:47/00:02:55, flags: sT
Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
Outgoing interface list:
  Tunnel0, Forward/Sparse, 00:00:47/00:02:55
```

PE1 e PE2 hanno entrambi ricevuto un'unione PIM per (10.100.1.6,232.1.1.1). L'interfaccia Tunnel0 è inclusa nell'elenco delle interfacce in uscita (OIL) per la voce multicast su entrambi i router.

Il traffico multicast inizia a scorrere per (10.100.1.6,232.1.1.1). "Debug ip pim vrf one 232.1.1.1" e "debug ip mrouting vrf one 232.1.1.1" mostrano che l'arrivo del traffico multicast sul tunnel0 (nell'OIL) di entrambi i router PE in entrata causa l'esecuzione del meccanismo di asserzione.

## PE1

```
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
MRT(1): not RPF interface, source address 10.100.1.6, group address 232.1.1.1
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.2
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We lose, our metric [110/11]
PIM(1): Prune Tunnel0/232.10.10.10 from (10.100.1.6/32, 232.1.1.1)
MRT(1): Delete Tunnel0/232.10.10.10 from the olist of (10.100.1.6, 232.1.1.1)
MRT(1): Reset the PIM interest flag for (10.100.1.6, 232.1.1.1)
MRT(1): set min mtu for (10.100.1.6, 232.1.1.1) 1500->18010 - deleted
PIM(1): Received v2 Join/Prune on Tunnel0 from 10.100.1.3, not to us
PIM(1): Join-list: (10.100.1.6/32, 232.1.1.1), S-bit set
```

## PE2

```
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.1
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We win, our metric [110/11]
PIM(1): (10.100.1.6/32, 232.1.1.1) oif Tunnel0 in Forward state
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): Received v2 Join/Prune on Tunnel0 from 10.100.1.3, to us
PIM(1): Join-list: (10.100.1.6/32, 232.1.1.1), S-bit set
PIM(1): Update Tunnel0/10.100.1.3 to (10.100.1.6, 232.1.1.1), Forward state, by PIM SG Join
```

Se la metrica e la distanza sono le stesse di entrambi i router verso l'origine 10.100.1.6, allora c'è un tie-breaker per determinare il vincitore dell'asserzione. L'interruttore differenziale è l'indirizzo IP più alto del router adiacente PIM sul tunnel0 (MDT predefinito). In questo caso, si tratta di PE2:

```
PE1#show ip pim vrf one neighbor
```

```
PIM Neighbor Table
```

```
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
```

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
10.100.1.4	Tunnel0	06:27:57/00:01:29	v2	1 / DR S P G
10.100.1.3	Tunnel0	06:28:56/00:01:24	v2	1 / S P G
10.100.1.2	Tunnel0	06:29:00/00:01:41	v2	1 / S P G

```
PE1#show ip pim vrf one interface
```

Address	Interface	Ver/ Mode	Nbr Count	Query Intvl	DR Prior	DR
---------	-----------	-----------	-----------	-------------	----------	----

10.2.1.1	Ethernet0/0	v2/S	0	30	1	10.2.1.1
10.2.4.1	Ethernet1/0	v2/S	0	30	1	10.2.4.1
10.100.1.1	Lspvif1	v2/S	0	30	1	10.100.1.1
10.100.1.1	Tunnel0	v2/S	3	30	1	10.100.1.4

PE1 ha rimosso Tunnel0 dall'OIL della voce multicast a causa delle asserzioni. Da quando l'OIL è diventato vuoto, la voce multicast è potata.

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:17:24/00:00:01, flags: sPT
Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
Outgoing interface list: Null
```

In PE2 il flag A è impostato sull'interfaccia Tunnel0, in quanto risulta vincitore.

```
PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:17:20/00:02:54, flags: sT
Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
Outgoing interface list:
  Tunnel0, Forward/Sparse, 00:17:20/00:02:54, A
```

PE2 invia periodicamente un'asserzione su Tunnel0 (MDT predefinito), immediatamente prima della scadenza del timer di asserzione. Pertanto PE2 rimane il vincitore assoluto.

```
PE2#
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
```

## Conclusioni

Il meccanismo di asserzione funziona anche con un'interfaccia tunnel nell'OIL. Le asserzioni vengono scambiate sul MDT predefinito quando i router PE in ingresso ricevono traffico multicast C-(S,G) sull'interfaccia del tunnel associato che si trova in OIL.

## Meccanismo di asserzione con MDT dati

Nella maggior parte dei casi, quando sono configurati gli MDT di dati, il meccanismo di asserzione viene comunque eseguito sull'MDT predefinito, in quanto il traffico C-(S,G) viene trasferito dall'MDT predefinito agli MDT di dati solo dopo tre secondi. In questo caso si verifica la stessa situazione descritta in precedenza. Si noti che **esiste solo un'interfaccia tunnel per ogni VRF abilitata per il multicast**: L'MDT predefinito e tutti gli MDT dati utilizzano solo un'interfaccia tunnel. Questa interfaccia tunnel viene utilizzata nell'OIL sui router PE in entrata o come interfaccia RPF sui router PE in uscita.

In alcuni casi è possibile che il meccanismo di asserzione non venga attivato prima della segnalazione dei Data MDT. È quindi possibile che il traffico multicast C-(S,G) inizi a essere inoltrato su un MDT dati su entrambi i router PE in ingresso PE1 e PE2. In questi casi, ciò potrebbe portare a un traffico multicast C-(S,G) duplicato permanente attraverso la rete principale MPLS. Per evitare ciò, è stata implementata questa soluzione: quando un router PE in ingresso vede un altro router PE in ingresso annunciare un MDT dati per il quale il router PE è anche un router PE in ingresso, si unisce a tale MDT dati. In linea di principio, solo i router PE in uscita (che dispongono di un ricevitore a valle) si unirebbero all'MDT dati. Poiché i router PE in ingresso si uniscono all'MDT dati annunciato da altri router PE in ingresso, il router PE in ingresso riceve il traffico multicast dall'interfaccia tunnel presente nell'OIL e pertanto attiva il meccanismo di asserzione e induce uno dei router PE in ingresso a interrompere l'inoltro del traffico multicast C-(S,G) sul relativo MDT dati (con l'interfaccia tunnel), mentre l'altro router PE in ingresso (il vincitore dell'asserzione) può continuare a inoltrare il traffico multicast C-(S,G) sul relativo MDT dati.

Nell'esempio successivo si supponga che i router PE in ingresso PE1 e PE2 non abbiano mai visto il traffico multicast C-(S,G) l'uno dall'altro sull'MDT predefinito. Il traffico viene registrato sull'MDT predefinito solo per tre secondi e non è difficile capire che ciò possa verificarsi in caso, ad esempio, di una perdita temporanea del traffico sulla rete principale.

La configurazione per Data MDT viene aggiunta a tutti i router PE. La configurazione su tutti i router PE (il RD può essere diverso sui router PE) è:

```
vrf definition one
 rd 1:1
 !
 address-family ipv4
  mdt default 232.10.10.10
  mdt data 232.11.11.0 0.0.0.0
  route-target export 1:1
  route-target import 1:1
 exit-address-family
 !
```

Non appena PE1 e PE2 vedono il traffico proveniente dall'origine, creano una voce C-(S,G). Entrambi i router PE in ingresso inoltrano il traffico multicast C-(S,G) sull'MDT predefinito. I router PE in uscita PE3 e PE4 ricevono il traffico multicast e lo inoltrano. A causa di un problema temporaneo, PE2 non visualizza il traffico proveniente da PE1 e viceversa sull'MDT predefinito.

Entrambi inviano un valore TLV (Data MDT Join Type Length Value) sull'MDT predefinito.

Se non è presente traffico C-(S,G), sui router PE in ingresso verrà visualizzato questo stato multicast:

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:00:45/00:02:44, flags: sT
Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
Outgoing interface list:
  Tunnel0, Forward/Sparse, 00:00:45/00:02:42
```

```
PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:02:18/00:03:28, flags: sT
Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
Outgoing interface list:
  Tunnel0, Forward/Sparse, 00:02:18/00:03:28
```

Flag y non ancora impostato. Entrambi i router PE in ingresso hanno l'interfaccia Tunnel0 in OIL. Ciò è dovuto al fatto che PE3 ha RPF verso PE1 e PE4 ha RPF verso PE2 per C-(S,G).

Quando inizia il flusso del traffico multicast per C-(S,G), sia PE1 che PE2 inoltrano il traffico. La soglia per l'MDT dei dati viene superata su entrambi i router PE in ingresso ed entrambi inviano un TLV di join dell'MDT dei dati e dopo tre secondi iniziano l'inoltro sull'MDT dei dati. Si noti che PE1 si unisce all'MDT dati originato da PE2 e PE2 si unisce all'MDT dati originato da PE1.



```

PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:01:26/00:03:02, flags: sTy
Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
Outgoing interface list:
  Tunnel0, Forward/Sparse, 00:01:26/00:03:02

```

```

PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:00:41/00:02:48, flags: sTy
Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
Outgoing interface list:
  Tunnel0, Forward/Sparse, 00:00:41/00:02:48

```

Sia PE1 che PE ricevono traffico per C-(S,G) sull'interfaccia Tunnel0 (ma ora dal Data MDT, non dal Default MDT) e il meccanismo di asserzione viene attivato. Solo PE2 continua a inoltrare il traffico C-(S,G) sul proprio Data MDT:

```

PE1#
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
MRT(1): not RPF interface, source address 10.100.1.6, group address 232.1.1.1
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.2
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We lose, our metric [110/11]
PIM(1): Prune Tunnel0/232.11.11.0 from (10.100.1.6/32, 232.1.1.1)
MRT(1): Delete Tunnel0/232.11.11.0 from the olist of (10.100.1.6, 232.1.1.1)
MRT(1): Reset the PIM interest flag for (10.100.1.6, 232.1.1.1)
PIM(1): MDT Tunnel0 removed from (10.100.1.6,232.1.1.1)
MRT(1): Reset the y-flag for (10.100.1.6,232.1.1.1)

```

```
PIM(1): MDT next_hop change from: 232.11.11.0 to 232.10.10.10 for (10.100.1.6, 232.1.1.1)
Tunnel0
MRT(1): set min mtu for (10.100.1.6, 232.1.1.1) 1500->18010 - deleted
PIM(1): MDT threshold dropped for (10.100.1.6,232.1.1.1)
PIM(1): Receive MDT Packet (9889) from 10.100.1.2 (Tunnel0), length (ip: 44, udp: 24), ttl: 1
PIM(1): TLV type: 1 length: 16 MDT Packet length: 16
```

```
PE2#
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.1
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We win, our metric [110/11]
PIM(1): (10.100.1.6/32, 232.1.1.1) oif Tunnel0 in Forward state
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PE2#
PIM(1): Received v2 Join/Prune on Tunnel0 from 10.100.1.3, to us
PIM(1): Join-list: (10.100.1.6/32, 232.1.1.1), S-bit set
PIM(1): Update Tunnel0/10.100.1.3 to (10.100.1.6, 232.1.1.1), Forward state, by PIM SG Join
MRT(1): Update Tunnel0/232.10.10.10 in the olist of (10.100.1.6, 232.1.1.1), Forward state - MAC
built
MRT(1): Set the y-flag for (10.100.1.6,232.1.1.1)
PIM(1): MDT next_hop change from: 232.10.10.10 to 232.11.11.0 for (10.100.1.6, 232.1.1.1)
Tunnel0
```

**PE1 non dispone più dell'interfaccia tunnel in OIL.**

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.100.1.6, 232.1.1.1), 00:10:23/00:00:04, flags: sPT
Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
Outgoing interface list: Null
```

**In PE2 il flag A è impostato sull'interfaccia Tunnel0:**

```
PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
```

V - RD & Vector, v - Vector, p - PIM Joins on route,  
x - VxLAN group  
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join  
Timers: Uptime/Expires  
Interface state: Interface, Next-Hop or VCD, State/Mode  
  
(10.100.1.6, 232.1.1.1), 00:10:00/00:02:48, flags: sTy  
Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6  
Outgoing interface list:  
Tunnel0, Forward/Sparse, 00:08:40/00:02:48, A

## Conclusioni

Il meccanismo di asserzione funziona anche quando vengono utilizzati MDT di dati. Le asserzioni vengono scambiate sul MDT predefinito quando i router PE in ingresso ricevono traffico multicast C-(S,G) sull'interfaccia del tunnel associato che si trova in OIL.