

Load Balancing

Ericsson Core CLI

Operating Instructions

Copyright

© Ericsson AB 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.



Contents

1	Overview	1
1.1	Load Balancing for LERs and P Routers in MPLS Networks	1
1.2	Load Balancing in BGP Networks	2
1.3	Load Balancing Restrictions	4
1.4	Egress Optimization	5
2	Examples	6
2.1	Load-Balance an LER	6
2.2	Load-Balance an LER Using Static LSPs	8
2.3	Load-Balance a P Router (Transit Node)	17
2.4	Load-Balance a P Router with Two Next Hops	18
2.5	Load-Balance a BGP Router	21
2.6	Load-Balance a BGP Router with Two Next Hops	22
3	Load Balancing Reference List	25





1 Overview

1.1 Load Balancing for LERs and P Routers in MPLS Networks

By default, load-balancing algorithms (also known as hashing) are based on Layer 3 (L3) data, which includes only the source and destination IP addresses and the protocol type. This is also known as the 3-tuple hashing option. You can change the hashing to use Layer 4 data in the algorithm (the 5-tuple hashing option), which includes the source and destination IP addresses, the protocol type, and the source and destination ports for User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) streams.

Note: If you use Layer 4 hashing for the following packet types, the router automatically reverts to Layer 3 hashing:

- Non-UDP or TCP packets
- Packets with IP options
- Fragmented packets

If you configure the `service load-balance ip` command, the same load-balancing algorithm is applied for Equal-Cost Multipath (ECMP).

You can also enable Equal-Cost Multipath (ECMP) on P routers in which Label Distribution Protocol (LDP) is used for label distribution.

[Table 1](#) provides details of load balancing for traffic in LER configurations (based on *RFC 6178*).

Table 1 LER Load Balancing

Application	Traffic Type	Hashing
Plain IP	IP	IP source and destination and the protocol type. ⁽¹⁾
IP multi-cast		IP source and destination and the protocol type.
Generic Routing Encapsulation (GRE) with IP over IP	IP	Inner IP source and destination and the protocol type. ⁽¹⁾
IP Security (IPsec) tunnel	IP	Inner IP source and destination and the protocol type. ⁽¹⁾



Application	Traffic Type	Hashing
		If the IPsec tunnel is configured through link groups, the hashing is based on the IP source and destination and the protocol type.
IP-based MPLS (L3VPN)	IP	IP source and destination and the protocol type. ⁽¹⁾

(1) If the 5-tuple hashing option is configured, load balancing is based on the IP source, IP destination, protocol type, UDP or TCP source port, and UDP or TCP destination port.

Load-balancing hashing for egress LERs (eLERs) depends on the topology of the network into which packets are being forwarded. For example:

- Traffic forwarded as IP ECMP traffic is hashed based on Layer 3 information.
- Traffic forwarded into LSPs in another MPLS cloud (for example, through an L3VPN) is hashed based on the rules in [Table 1](#).

By default, on P routers, load balancing is based on hashing up to four labels of the label stack.

Optionally, the behavior of load balancing can be modified with the `service load-balance mpls-payload` command. With this command, the load balancing of ECMP and LAG is calculated from the MPLS payload. In the table below, the **Configured** state refers to the usage of this command. The **Not configured** state refers to the default behavior, or it also can be set with the `no service load-balance mpls-payload` command.

Note: On P routers, non-labeled IP traffic is either balanced by the default IP hashing (IP source and destination and the protocol type) or an algorithm configured using the `service load-balance ip` command.

1.2 Load Balancing in BGP Networks

You can configure load balancing in networks that use BGP route advertisement by using the BGP multi-path capabilities. By default, BGP multi-path is disabled, which means that BGP installs a single path in the Routing Information Base (RIB) for each destination. If that path fails and no other path has installed a path for that prefix, traffic destined for that path is lost until the path is available again.

When BGP multi-path is enabled, BGP installs multiple best equal-cost paths in the routing table to balance the traffic load to BGP destinations. You can configure one of the following multi-paths:

- All internal BGP (iBGP)



- All external BGP (eBGP)
- A combination of iBGP and eBGP in a VPN context, with only one eBGP path allowed

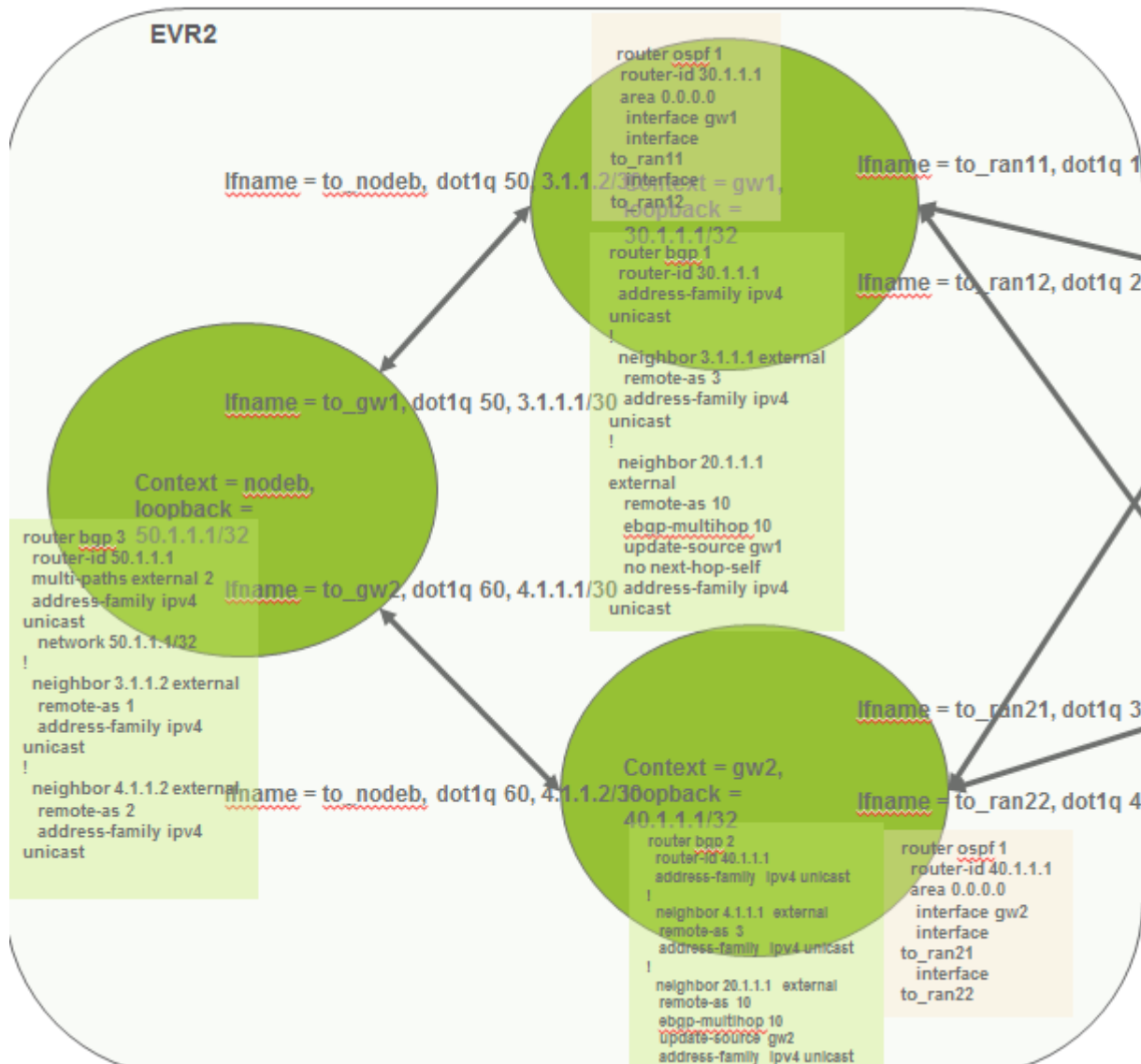
When BGP multi-path capabilities are enabled, even though multiple paths are installed in the RIB, BGP advertises only one path—the BGP best path—to its peers.

Load Balancing Using Hierarchical ECMP: Multipath BGP over OSPF

Hierarchical ECMP is a general capability provided by routing stack. IPOS routing stack can be used to achieve BGP based load balancing.

To enable BGP based load balancing, enable `multi-path` BGP from respective context. BGP can load balance up to 16 paths. Enabling multipath enables BGP to construct multiple available paths towards the same destination instead of picking one of the best available paths. The best available path with the same metric can be built using static or IGP based routes.

The following example illustrates sample scenario and configurations for hierarchical ECMP multipath BGP over OSPF:



With this configuration, BGP builds ECMP over the OSPF routes for every IP address reachable via one or more datacentre gateway (multi-path external 2 in the above example).

This solution involves an additional forwarding information base table lookup than the regular solution for every routed packet to external sources. With this configuration, we can support up to 16 links per datacentre gateway and up to 16 datacentre gateways.

1.3 Load Balancing Restrictions



Traffic can be balanced on up to 16 equal-cost links. If you configure more than 16 links, the excess links are not included in the load-balancing algorithms.

When `tunnel lsp ldp` is configured for port pseudowires or pseudowires, packet load-balancing across multiple equal-cost multipaths (ECMPs) does not take effect.

1.4 Egress Optimization

In virtual router environments, the capacity of the virtual fabric is a limiting factor in terms of line rate traffic. If the egress and ingress PFEs would be the same, egress optimization allows the packet to loop back and be processed within the same routing context. While in some special cases it may be beneficial to span traffic across multiple cards, providing application hosting and egress processing within a single VM yields a significant performance increase by eliminating the overhead of an extra hop to a vFRWD.



2 Examples

2.1 Load-Balance an LER

This example load-balances a router configured as an LER in an MPLS network.

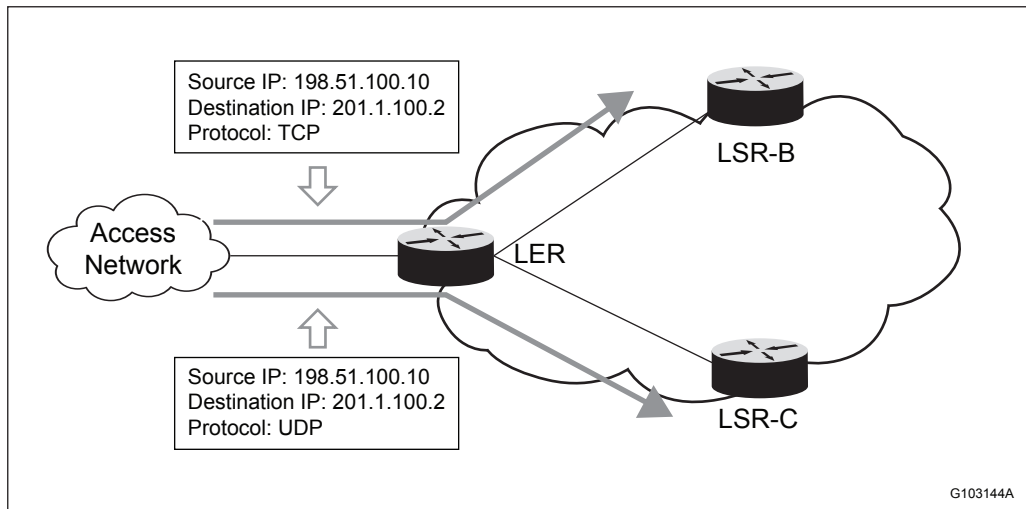


Figure 1 3-Tuple Load Balancing for an LER

Figure 1 shows the router balancing traffic load between LSR-B and LSR-C based on the 3-tuple (source IP, destination IP, protocol type) hashing option. You can also configure load balancing based on the 5-tuple (source IP, destination IP, protocol type, source port, and destination port) hashing option in a similar way, as shown in Figure 2.

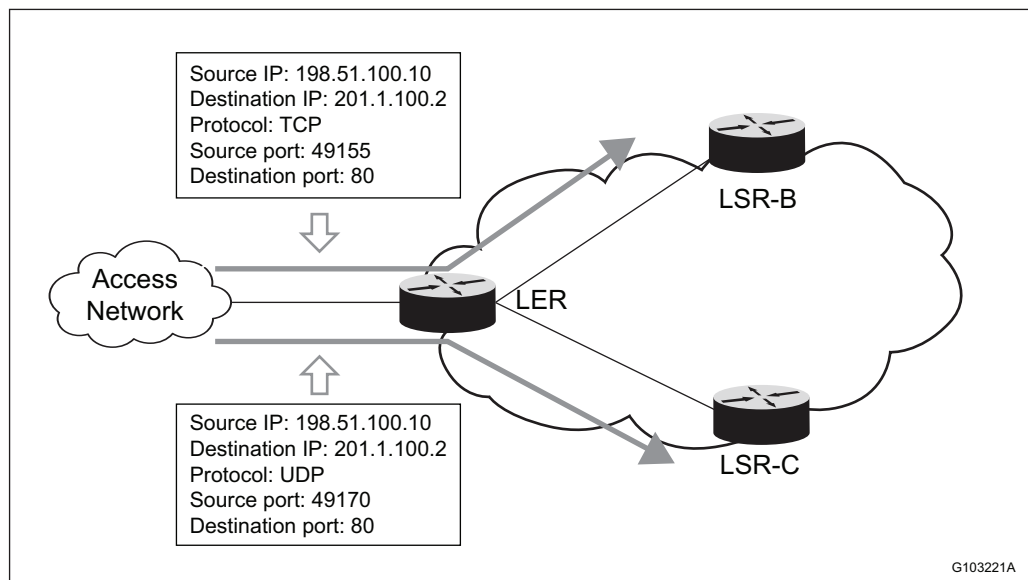


Figure 2 Hashing for Load Balancing

To load-balance the LER with 5-tuple hashing, do the following starting in global configuration mode.

Note: You can return hashing to the default 3-tuple hashing, using the **layer-3** keyword and source IP header, using the **source-only** keyword.

Steps

1. Configure load balancing to use the 5-tuple hashing option.

Example

```
Ericsson(config)#contexts service load-balance ip layer-4
Ericsson(config)#commit
Ericsson(config)#end
```

2. Confirm the configuration. The load-balance field shows **layer-4**.

Example

```
[local]Ericsson#show ip route summary
load-balance: layer-4
Rt Tbl Version:      67, Nh Tbl Version:    109
FIB Rt Tbl Version: 67
Route Source                               Tot-Routes    Act-Route →
s Max Ever Reached

Connected                                     3              →
3
Static                                       1              →
1
```



Mpls static 2 3 2 →

2.2 Load-Balance an LER Using Static LSPs

This example load-balances a router configured as an LER in an MPLS network using static LSPs.

- Subnet 198.51.100.0/24 → LER → LSR-B → LSR-D → Subnet 192.0.2.96/28
- Subnet 198.51.100.0/24 → LER → LSR-C → LSR-D → Subnet 192.0.2.96/28

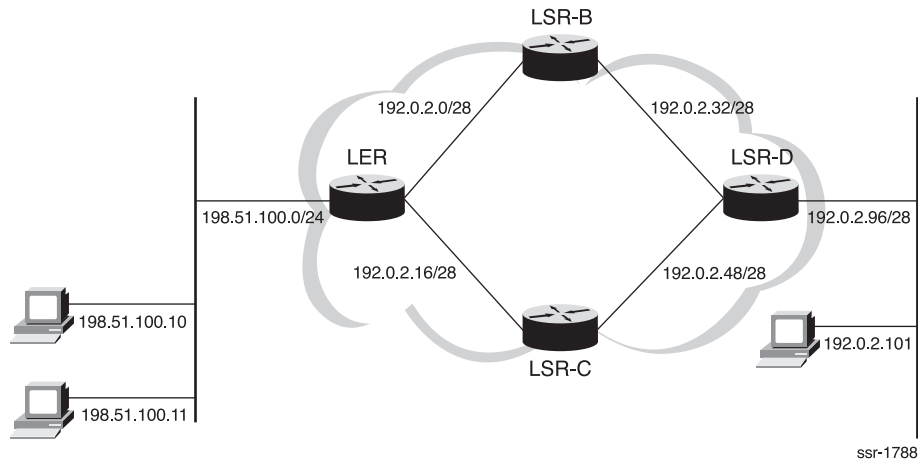


Figure 3 Load-Balanced LER in a Network

Figure 3 shows an MPLS network forwarding traffic between two subnets. The LER is configured with static LSPs that use the following labels between the inbound and outbound LSRs:

- LER → LSR-B → LSR-D uses label 16 between the ingress and LSR-B and label 46 to the egress on LSR-D.
- LER → LSR-C → LSR-D uses label 26 between the ingress and LSR-C and label 56 to the egress on LSR-D.
- LER ← LSR-B ← LSR-D uses label 46 between the ingress and LSR-B, then label 16 to the egress on LER.
- LER ← LSR-C ← LSR-D uses label 56 between the ingress and LSR-C, then label 26 to the egress on LER.

The LER balances traffic by default. This example demonstrates this load balancing by pinging a host in one subnet from different source addresses in another subnet and showing that the packets travel over either LSR-B or LSR-C, depending on the source address.



2.2.1 Configure the LER

In this example, the LER is connected to subnet 198.51.100.0/24 with interface `if2subnet` using IP address 198.51.100.1/24 to LSR-B with interface `if2B` using IP address 192.0.2.01/28, and to LSR-C with interface `if2C` using IP address 192.0.2.21/28. For information on configuring interfaces, see *Contexts and Interfaces*. The router is then configured with two static LSPs to LSR-D.

To configure the LER, do the following starting in global configuration mode.

Steps

1. Enter the context.

Example

```
LER(config)#contexts context local
```

2. Configure a static LSP to LSR-D through LSR-B.

Example

```
LER(config-context-local)#router mpls-static lsp DviaB
LER(config-lsp-DviaB)#egress 192.0.2.42
LER(config-lsp-DviaB)#next-hop 192.0.2.2
LER(config-lsp-DviaB)#out-label 16
LER(config-lsp-DviaB)#exit
```

3. Configure a static LSP to LSR-D through LSR-C.

Example

```
LER(config-context-local)#router mpls-static lsp DviaC
LER(config-lsp-DviaC)#egress 192.0.2.52
LER(config-lsp-DviaC)#next-hop 192.0.2.22
LER(config-lsp-DviaC)#out-label 26
LER(config-lsp-DviaC)#exit
```

4. Configure interfaces `if2B` and `if2C` as egresses for two LSPs from LSR-D.

Example

```
LER(config-context-local)#router mpls-static interface →
  if2B@local
LER(config-interface-if2B@local)#label-action 16
LER(config-label-action-16)#pop
LER(config-label-action-16)#exit
LER(config-interface-if2B@local)#exit
LER(config-context-local)#router mpls-static interface →
```



```
if2C@local
LER(config-interface-if2C@local)#label-action 26
LER(config-label-action-26)#pop
LER(config-label-action-26)#exit
LER(config-interface-if2C@local)#exit
```

5. Configure two static routes to the 192.0.2.96/28 subnet, one through LSR-B and one through LSR-C.

Example

```
LER(config-context-local)#ip static-route 192.0.2.96/2 →
8 next-hop-address 192.0.2.42
LER(config-static-route-192.0.2.96/28/next-hop-address →
/192.0.2.42)#exit
LER(config-context-local)#ip static-route 192.0.2.96/2 →
8 next-hop-address 192.0.2.52
LER(config-static-route-192.0.2.96/28/next-hop-address →
/192.0.2.52)#exit
LER(config-context-local)#exit
LER(config)#commit
LER(config)#exit
```

6. Use the `show running-config` command to confirm the configuration.

2.2.2

Configure LSR-B

In this example, router LSR-B is connected to the LER with interface `if2A` using IP address 192.0.2.02/28, and to LSR-D with interface `if2D` using IP address 192.0.2.41/28. The router is then configured to forward MPLS traffic between the LER and LSR-D.

To configure the LSR, do the following starting in global configuration mode.

Steps

1. Enter the local context.

Example

```
LSR-B(config)#contexts context local
```

2. Configure interfaces `if2A` and `if2D` for MPLS traffic between the LER and LSR-D.

Example

```
LSR-B(config-context-local)#router mpls-static interfa →
ce if2A@local
LSR-B(config-interface-if2A@local)#label-action 16
LSR-B(config-label-action-16)#swap outlbl-val 46
LSR-B(config-label-action-16)#swap nhop-addr 192.0.2.4 →
```



```

2
LSR-B (config-label-action-16) #exit
LSR-B (config-interface-if2A@local) #exit
LSR-B (config-context-local) #router mpls-static interfa →
ce if2D@local
LSR-B (config-interface-if2D@local) #label-action 46
LSR-B (config-label-action-46) #swap outlbl-val 16
LSR-B (config-label-action-46) #swap nhop-addr 192.0.2.1
LSR-B (config-label-action-46) #commit
LSR-B (config-label-action-46) #end

```

3. Use the `show running-config` command to confirm the configuration.

2.2.3 Configure LSR-C

In this example, router LSR-C is connected to the LER with interface `if2A` using IP address `192.0.2.22/28`, and to LSR-D with interface `if2D` using IP address `192.0.2.51/28`. The router is then configured to forward MPLS traffic between the LER and LSR-D.

Repeat the procedure used to configure LSR-B, as follows:

Steps

1. Configure interface `if2A` to swap ingress label `26` for egress label `56`.
2. Configure interface `if2D` to swap ingress label `56` for egress label `26`.

2.2.4 Configure LSR-D

In this example, router LSR-D is connected to subnet `192.0.2.96/28` with interface `if2subnet` using IP address `192.0.2.101/28`, to LSR-B with interface `if2B` using IP address `192.0.2.42/28`, and to LSR-C with interface `if2C` using IP address `192.0.2.52/28`.

Repeat the procedure used to configure the LER, as follows:

Steps

1. Configure a static LSP to the LER `AviaB` (LER ← LSR-B ← LSR-D), with egress `192.0.2.01`, next hop `192.0.2.41`, and label `46`.
2. Configure a static LSP to the LER `AviaC` (LER ← LSR-C ← LSR-D), with egress `192.0.2.21`, next hop `192.0.2.51`, and label `56`.
3. Configure two static routes for the `198.51.100.0/24` subnet, one through LSR-B and one through LSR-C.



2.2.5 Verify the Network Configuration

To confirm the configuration, do the following in operational mode.

Steps

1. On the LER, verify the two LSPs and two static routes to the 192.0.2.96/28 subnet.

Example

```
[local]LER#show ip route all
Codes: C - connected, S - static, S dv - dvsr, R - RIP →
, e B - EBGP, i B - IBGP
      O - OSPF, O3 - OSPFv3, IA - OSPF(v3) inter-a →
rea,
...
*S      192.0.2.42/32          192.0.2.2          5      →
  0          if2B
*S      192.0.2.52/32          192.0.2.22         5      →
  0          if2C
> S      192.0.2.96/28          192.0.2.42         1      →
  0 03:54:25 via lsp cct
>
                                192.0.2.52          →
                                via lsp cct
...
```

2. On the LER, verify that the two LSPs are up.

Example

```
[local]LER#show ip route lsp
Codes: S - static, RSVP - RSVP, LDP - LDP, BGP - BGP
      O - OSPF, TSC - tunnel shortcut
      > - Lowest distance path
Type   Network                UpTime Interface      La →
bel    LSP Circuit
> S    192.0.2.42/32          1d02h if2B             1 →
6      255/3:511:63:31/0/1/3
> S    192.0.2.52/32          1d02h if2C             2 →
6      255/3:511:63:31/0/1/4
```

3. On LSR-B, verify that the two LSPs are up.

Example

```
[local]LSR-B#show mpls-static label-action detail

In-label          : 16          Action          : →
swap
```



```

Out-label          : 46          Nhop          : →
  192.0.2.42
State              : Up
Outgoing circuit   : 1/2:511:63:31/1/1/7
Outgoing intf grid : 0x10000002

In-label          : 46          Action         : →
  swap
Out-label         : 16          Nhop          : →
  192.0.2.1
State             : Up
Outgoing circuit   : 1/1:511:63:31/1/1/5
Outgoing intf grid : 0x10000001

```

4. On LSR-C, verify that the two LSPs are up.

Example

```
[local]LSR-C#show mpls-static label-action detail
```

```

In-label          : 26          Action         : →
  swap
Out-label         : 56          Nhop          : →
  192.0.2.52
State             : Up
Outgoing circuit   : 1/2:511:63:31/1/1/7
Outgoing intf grid : 0x10000002

In-label          : 56          Action         : →
  swap
Out-label         : 26          Nhop          : →
  192.0.2.21
State             : Up
Outgoing circuit   : 1/1:511:63:31/1/1/5
Outgoing intf grid : 0x10000001

```

5. On LSR-D, verify the two LSPs and two static routes to the 198.51.100.10/24 subnet.

Example

```
[local]LSR-D#show ip route all
```

```

Codes: C - connected, S - static, S dv - dvsr, R - RIP →
, e B - EBGp, i B - IBGP
      O - OSPF, O3 - OSPFv3, IA - OSPF(v3) inter-a →
rea,
...
*S    192.0.2.1/32          192.0.2.41          5          →
  0          if2B
*S    192.0.2.21/32        192.0.2.51          5          →
  0          if2C

```



```

...
> S      198.51.100.0/24      192.0.2.1      1      →
   0 04:07:55 via lsp cct
>
   192.0.2.21      →
   via lsp cct
...

```

6. On LSR-D, verify that the two LSPs are up.

Example

```

[local]LSR-D#show ip route lsp
Codes: S - static, RSVP - RSVP, LDP - LDP, BGP - BGP
       O - OSPF, TSC - tunnel shortcut
       > - Lowest distance path
Type   Network                UpTime Interface      La →
bel    LSP Circuit
> S    192.0.2.1/32          05:21:34 if2B          46 →
       255/3:511:63:31/0/1/3
> S    192.0.2.21/32       05:21:34 if2C          56 →
       255/3:511:63:31/0/1/4

```

2.2.6 Verify Load Balancing

To verify load balancing, this example pings a host in a subnet connected to LSR-D from a subnet connected to the LER and verifies that the traffic from different source addresses is hashed to different next hops, in this case, either LSR-B or LSR-C.

To verify load balancing, do the following in operational mode.

Steps

1. On LSR-B and LSR-C, clear the port counters.

Example

```

[local]LSR-B#clear port counters
[local]LSR-B#show port counters live
please wait...
Port      Type
1/1       ethernet
packets sent      : 0          bytes sent      →
: 0
packets recvd     : 0          bytes recvd     →
: 0
send packet rate  : 0.00       send bit rate   →
: 0.00
recv packet rate  : 0.00       recv bit rate   →
: 0.00
rate refresh interval : 60 seconds

```



```

1/2          ethernet
packets sent      : 0          bytes sent      →
      : 0
packets recvd     : 0          bytes recvd     →
      : 0
send packet rate  : 0.00       send bit rate   →
      : 0.00
recv packet rate  : 0.00       recv bit rate   →
      : 0.00
rate refresh interval : 60 seconds
...

```

2. Ping the host with IP address **192.0.2.101/28** from the host with IP address **198.51.100.11**.

Example

```

[local]Host1#ping 192.0.2.101
PING 192.0.2.101 (192.0.2.101): source 198.51.100.11, →
36 data bytes,
timeout is 1 second
!!!!!!

----192.0.2.101 PING Statistics----
5 packets transmitted, 5 packets received, 0.0% packet →
loss
round-trip min/avg/max/stddev = 10.531/12.752/18.731/3 →
.392 ms

```

3. On LSR-B and LSR-C, check the port counters. LSR-B is shown here. There is no change to LSR-C. Note the increment in the values displayed in the **packets sent** and **packets recvd** fields.

Example

```

[local]LSR-B#show port counters live
please wait...
Port          Type
1/1           ethernet
packets sent   : 5          bytes sent     →
      : 820
packets recvd  : 5          bytes recvd    →
      : 410
send packet rate : 0.00       send bit rate  →
      : 0.00
recv packet rate : 0.00       recv bit rate  →
      : 0.00
rate refresh interval : 60 seconds

1/2          ethernet

```



```
packets sent      : 5          bytes sent      →
   : 820
packets recvd    : 5          bytes recvd    →
   : 410
send packet rate : 0.00      send bit rate  →
   : 0.00
recv packet rate : 0.00      recv bit rate  →
   : 0.00
rate refresh interval : 60 seconds

management      ethernet
packets sent     : 0          bytes sent     →
   : 0
packets recvd   : 0          bytes recvd    →
   : 0
...
```

4. Ping the host with IP address **192.0.2.101/28** from the host with IP address **198.51.100.10**. In this example, this is done by changing the IP address of the interface connected to the LER.

Example

```
[local]Host1#config
Host1(config)#interfaces interface ifEthernet1@local t →
ype ipForward l3-interface context local
Host1(config-interface-ifEthernet1@local)#l3-interface →
ip address addr-primary addr 198.51.100.10/24
Host1(config-interface-ifEthernet1@local)#commit
Host1(config-interface-ifEthernet1@local)#end
[local]Host1#ping 192.0.2.101
PING 192.0.2.101 (192.0.2.101): source 198.51.100.10, →
36 data bytes,
timeout is 1 second
!!!!!!

----192.0.2.101 PING Statistics----
5 packets transmitted, 5 packets received, 0.0% packet →
loss
round-trip min/avg/max/stddev = 8.679/11.545/18.437/3. →
979 ms
```

5. On LSR-B and LSR-C, check the port counters. LSR-C is shown here. There is no change to LSR-B. Note the increment in the values displayed in the **packets sent** and **packets recvd** fields.

Example

```
[local]LSR-C#show port counters live
please wait...
Port                Type
```

```

1/1          ethernet
packets sent      : 5          bytes sent      →
      : 820
packets recvd    : 5          bytes recvd    →
      : 410
send packet rate : 0.00      send bit rate  →
      : 0.00
recv packet rate : 0.00      recv bit rate  →
      : 0.00
rate refresh interval : 60 seconds

1/2          ethernet
packets sent      : 6          bytes sent      →
      : 904
packets recvd    : 6          bytes recvd    →
      : 470
send packet rate : 0.00      send bit rate  →
      : 0.00
recv packet rate : 0.00      recv bit rate  →
      : 0.00
rate refresh interval : 60 seconds

management      ethernet
packets sent      : 0          bytes sent      →
      : 0
packets recvd    : 0          bytes recvd    →
      : 0
...

```

2.3 Load-Balance a P Router (Transit Node)

This example load-balances a router configured as a P router in an MPLS network.

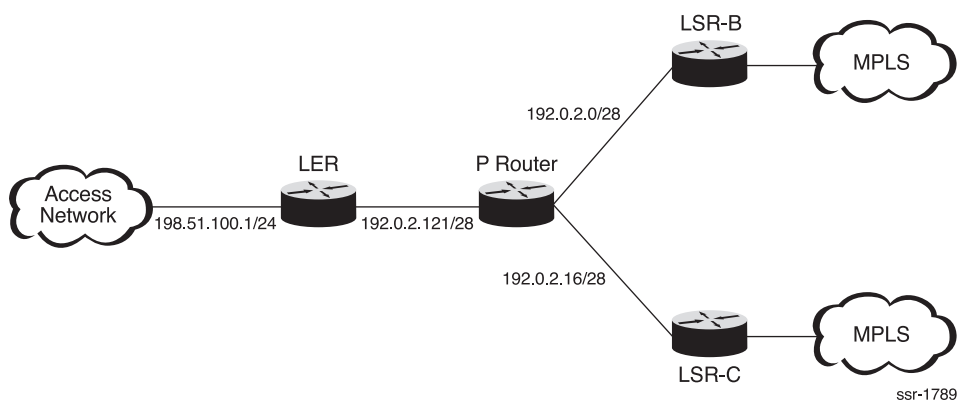


Figure 4 Load Balancing for a P router

Figure 4 shows the router using ECMP to balance the traffic load between LSR-B and LSR-C.



You can also use the `service load-balance mpls payload` command to configure load balancing based on the MPLS payload.

To load-balance a P router, do the following starting in global configuration mode.

Steps

1. Enter the context, access LDP configuration mode, and enable ECMP.

Example

```
Ericsson(config)#contexts context local
Ericsson(config-context-local)#router ldp ecmp-transit
Ericsson(config-context-local)#exit
Ericsson(config)#commit
Ericsson(config)#exit
```

2. Use the `show running-config` command to confirm the configuration.

2.4 Load-Balance a P Router with Two Next Hops

This example load-balances traffic from router LSR-A, configured as a P router in an MPLS network, across two next hops to an egress router.

- LER → LSR-A → LSR-B → LSR-D → Subnet 192.0.2.96/28
- LER → LSR-A → LSR-C → LSR-D → Subnet 192.0.2.96/28

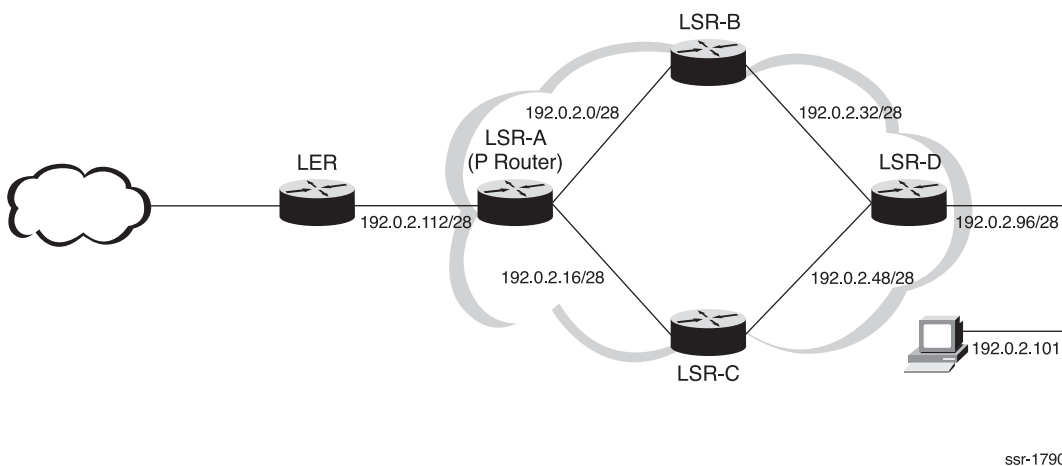


Figure 5 Load-Balanced P Router in a Network

Figure 5 shows an MPLS network forwarding traffic between two subnets configured as follows:

- LSR-A links to the LER with interface `if2LER` using IP address `192.0.2.121/28`.



- LSR-A links to LSR-B with interface `if2B` using IP address `192.0.2.01/28`.
- LSR-A links to LSR-C with interface `if2C` using IP address `192.0.2.21/28`.
- LSR-D links to LSR-B over subnet `192.0.2.32/28` and to LSR-C over subnet `192.0.2.48/28`. It is connected to subnet `192.0.2.96/28`.

Open Shortest Path First (OSPF) is the underlying Interior Gateway Protocol (IGP), and LDP distributes labels for MPLS. This example demonstrates load balancing on P router LSR-A and shows that when ECMP is enabled, MPLS provides paths over both LSR-B and LSR-C.

2.4.1 Configure the P Router

To configure the P router, do the following starting in global configuration mode.

Steps

1. Enter the context, access LDP configuration mode, and enable ECMP.

Example

```
LSR-A(config)#contexts context local
LSR-A(config-context-local)#router ldp ecmp-transit
LSR-A(config-context-local)#exit
LSR-A(config)#commit
LSR-A(config)#exit
```

2. Use the `show running-config` command to confirm the configuration.

2.4.2 Verify Load Balancing

To verify load balancing, this example uses the `traceroute mpls ldp` command to show that MPLS provides paths over either LSR-B or LSR-C.

To verify load balancing, do the following in operational mode.

Steps

1. Send packets to IP address `192.0.2.101/28` from the LER. Hop 1 has two available next hops: `192.0.2.2` (LSR-B) and `192.0.2.22` (LSR-C). The packets travel through LSR-C.

Example

```
[local]LER#traceroute mpls ldp 192.0.2.101/28 verbose
Sending 1 100-byte MPLS echos to LDP 192.0.2.101/28, s →
```



```

source 192.0.2.122,
    timeout is 1 second, send interval is 0 msec:
Pkt Hop      Recv Intf      Down Intf      Down Nb →
r  RTT  MTU RC
  1    0      N/A      192.0.2.121    192.0.2.12 →
1    0  1500 Ingress
                                Out Label: →
655365
  1    1      192.0.2.121    192.0.2.2      192.0.2. →
2    8  1500 Transit
                                Recv Label: 655365      Out Label: →
655365
                                192.0.2.22      192.0.2.2 →
2                                1500
                                Recv Label: 655365      Out Label: →
655365
  1    2      192.0.2.22      192.0.2.52      192.0.2.5 →
2   10  1500 Transit
                                Recv Label: 655365      Out Label: →
                                3
  1    3      192.0.2.52      N/A              N/ →
A   11  N/A Egress

```

---- MPLS Traceroute Statistics----

```

1 path queries, 3 packets transmitted, 3 packets received no error
0.0% packet loss/error

```

2. Disable ECMP on the P router.

Example

```

LSR-A(config)#contexts context local
LSR-A(config-context-local)#no router ldp ecmp-transit
LSR-A(config-context-local)#commit
LSR-A(config-context-local)#end

```

3. Send packets again to IP address 192.0.2.101/28 from the LER. Hop 1 now has only 192.0.2.2 (LSR-B) available as its next hop. Re-enable ECMP on the P router before proceeding with your network configuration.

Example

```

[local]LER#traceroute mpls ldp 192.0.2.101/28 verbose
Sending 1 100-byte MPLS echos to LDP 192.0.2.101/28, source 192.0.2.124,
    timeout is 1 second, send interval is 0 msec:
Pkt Hop      Recv Intf      Down Intf      Down Nb →
r  RTT  MTU RC
  1    0      N/A      192.0.2.121    192.0.2.12 →
1    0  1500 Ingress

```

```

655365
 1 1 192.0.2.121 192.0.2.2 192.0.2. →
 2 8 1500 Transit Recv Label: 655365 Out Label: →
655365
 1 2 192.0.2.2 192.0.2.42 192.0.2.4 →
 2 5 1500 Transit Recv Label: 655365 Out Label: →
 3
 1 3 192.0.2.42 N/A N/ →
A 11 N/A Egress

```

----- MPLS Traceroute Statistics-----
1 path queries, 3 packet

2.5 Load-Balance a BGP Router

This example load-balances a BGP router connecting the networks of Autonomous System Numbers (ASNs) 1 and 2.

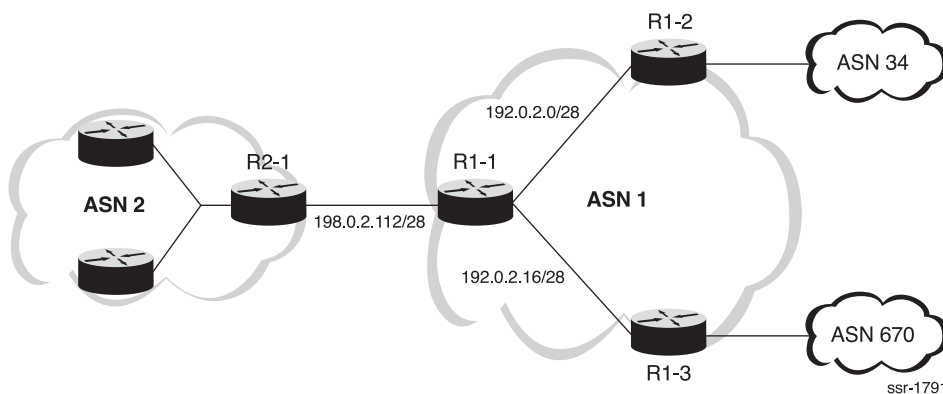


Figure 6 Load Balancing for a BGP Router

Figure 6 shows router R1-1 balancing the traffic load between R1-2 and R1-3 based on equal-cost iBGP paths. You can configure load balancing over equal-cost paths from different eBGP neighbors or a combination of internal and external paths. For information on BGP, see *BGP*.

To load-balance a BGP router, do the following starting in global configuration mode.

Steps

1. Enter the context, access BGP configuration mode, and configure multiple paths.



Example

```

R1-1 (config) #contexts context local
R1-1 (config-context-local) #router bgp 1
R1-1 (config-bgp-1) #multi-paths external internal 2
R1-1 (config-bgp-1) #exit
R1-1 (config-context-local) #exit
R1-1 (config) #commit
R1-1 (config) #exit

```

2. Use the `show running-config` command to confirm the configuration.

2.6 Load-Balance a BGP Router with Two Next Hops

This example load-balances traffic from BGP router R1-1 across two next hops to an egress router.

- Subnet 203.0.113.0/24 → R100-1 → R1-1 → R1-2 → R1-4 → Subnet 198.51.100.10/24
- Subnet 203.0.113.0/24 → R100-1 → R1-1 → R1-3 → R1-4 → Subnet 198.51.100.10/24

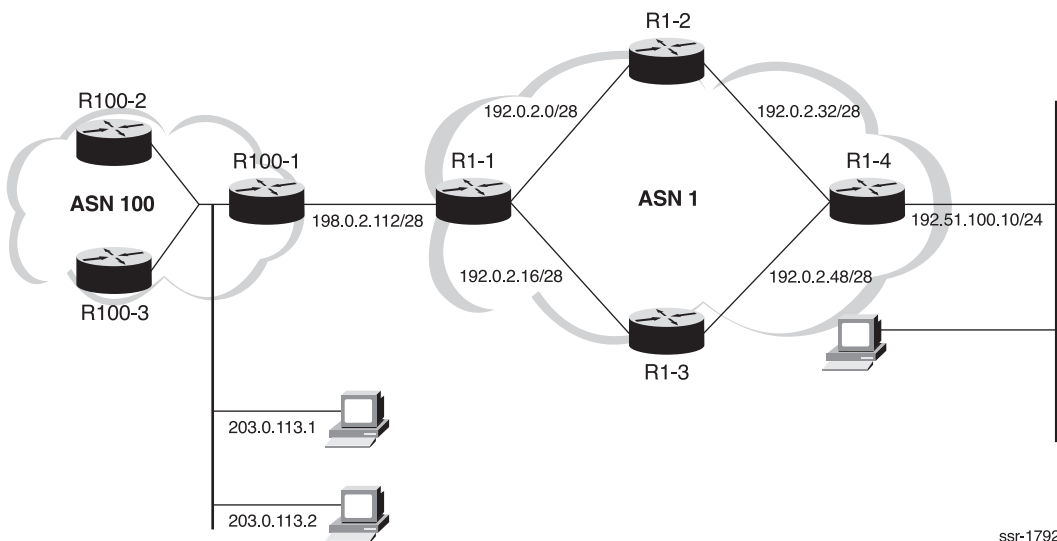


Figure 7 Load-Balanced BGP Router in a Network

Figure 7 shows BGP router R1-1 balancing incoming traffic from ASN 100 across R1-2 and R1-3. ASN 1 uses OSPF as the underlying IGP and BGP redistributes the path to subnet 198.51.100.10/24 to ASN 100.

This example demonstrates load balancing on the BGP router by sending packets from two IP addresses in a subnet connected to ASN 100, through R1-1, to a host in a subnet connected to ASN 1. BGP routes the packets over both R1-2 and R1-3.



2.6.1 Configure BGP Router R1-1

In this example, R1-1 is connected to R100-1 in ASN 100 with interface `if2A` using IP address `192.0.2.121/28` to R1-2 with interface `if2B` using IP address `192.0.2.01/28` and to R1-3 with interface `if2C` using IP address `192.0.2.21/28`.

The router is then configured to install multiple best equal-cost paths in the routing table for load-balancing traffic to BGP destinations.

To configure the BGP router, do the following starting in global configuration mode.

Steps

1. Enter the context, access BGP configuration mode, and configure multiple paths.

Example

```
R1-1 (config) #contexts context local
R1-1 (config-context-local) #router bgp 1
R1-1 (config-bgp-1) #multi-paths external internal 3
R1-1 (config-bgp-1) #exit
R1-1 (config-context-local) #exit
R1-1 (config) #commit
R1-1 (config) #exit
```

2. Use the `show running-config` command to confirm the configuration.

2.6.2 Verify Load Balancing

Verify that the network passes traffic over either R1-2 or R1-3 using the traceroute utility.

To verify load balancing, do the following in operational mode.

Steps

1. Send packets to IP address `198.51.100.0` from two source addresses in ASN 100. R100-1 represents hosts `203.0.113.2` and `203.0.113.1`. Hop 2 depends on the source address: R1-2 (`192.0.2.2`) for the first host, and R1-3 (`192.0.2.22`) for the second.

Example

```
[local]R100-1#traceroute 198.51.100.0 source 203.0.113 →
.2
traceroute to 198.51.100.0 (198.51.100.0), 30 hops max →
, 60 byte packets
 1 192.0.2.121 (192.0.2.121) 36.995 ms 36.640 ms 3 →
```



```
7.101 ms
 2 192.0.2.2 (192.0.2.2) 37.221 ms 37.044 ms 36.86 →
9 ms
 3 198.51.100.0 (198.51.100.0) 36.649 ms 37.397 ms →
37.223 ms

[local]R100-1#traceroute 198.51.100.0 source 203.0.113 →
.1
traceroute to 198.51.100.0 (198.51.100.0), 30 hops max →
, 60 byte packets
 1 192.0.2.121 (192.0.2.121) 36.007 ms 37.914 ms 3 →
7.683 ms
 2 192.0.2.22 (192.0.2.22) 37.446 ms 38.243 ms 35. →
381 ms
 3 198.51.100.0 (198.51.100.0) 34.699 ms 34.949 ms →
34.766 ms
```



3 Load Balancing Reference List

RFCs

- *RFC 6178, Label Edge Router Forwarding of IPv4 Option Packets*
- *RFC 4364, BGP/MPLS IP Virtual Private Networks (VPNs)*
- *Ericsson IP Operating System Architecture, 3/155 53-CRA 119 2158-V6 Uen H*