

MPLS 流量工程中的路由解析

Juniper 网络公司，爱立信公司，2000 年 11 月

目录

内容提要.....	1
演示网络概述.....	2
路由信息数据库.....	4
使用 TRAFFIC-ENGINEERING BGP	7
使用 IGP 捷径.....	11
使用 TRAFFIC-ENGINEERING BGP-IGP.....	17
安装单一前缀.....	23
附录 A.....	26
RTR1	26
RTR2	27
RTR3	28
RTR4	29
RTR5	30
RTR6	30
RTR7	31
缩略语.....	32

内容提要

多协议标记交换 (MPLS) 流量工程把特定的数据流量映射到已经建立的标记交换路径(LSP)上，而不是映射到IGP计算的作为最佳(最短)路径一部分的数据链路上。对这一功能来说，最重要的一点是确定哪些流量将映射到LSP上。

通过使用特定的信宿前缀把出口LSR指定为下站路由器，流量被映射到隧道入口标记交换路由器 (LSR) 上的一个LSP上。该LSP并没有构成到信宿的一条整个路由，理解这一点非常

重要。相反，该LSP是路由的一个下站网段。因此，如果在路由解析过程中，出口LSR被认为是一个可行的下站备选路由器，那么只能把分组映射到一个LSP上。本文解释了Juniper Networks JUNOS Internet软件提供的各种选项，以在路由解析过程中包括LSP。

本文假设您已经了解了MPLS和JUNOS命令行界面的基本知识。

演示网络概述

所有路由解析实例都采用相同的网络拓扑结构。本节提供了理解后续章节中的实例必需的参考图。

图1说明了网络中使用的路由器、路由器名称、物理连接和IP地址。共有两个自治系统和七台路由器，即RTR1到RTR7。环回接口地址作为OSPF和BGP路由器ID，在适用时则作为IBGP和LSP连接的端点。

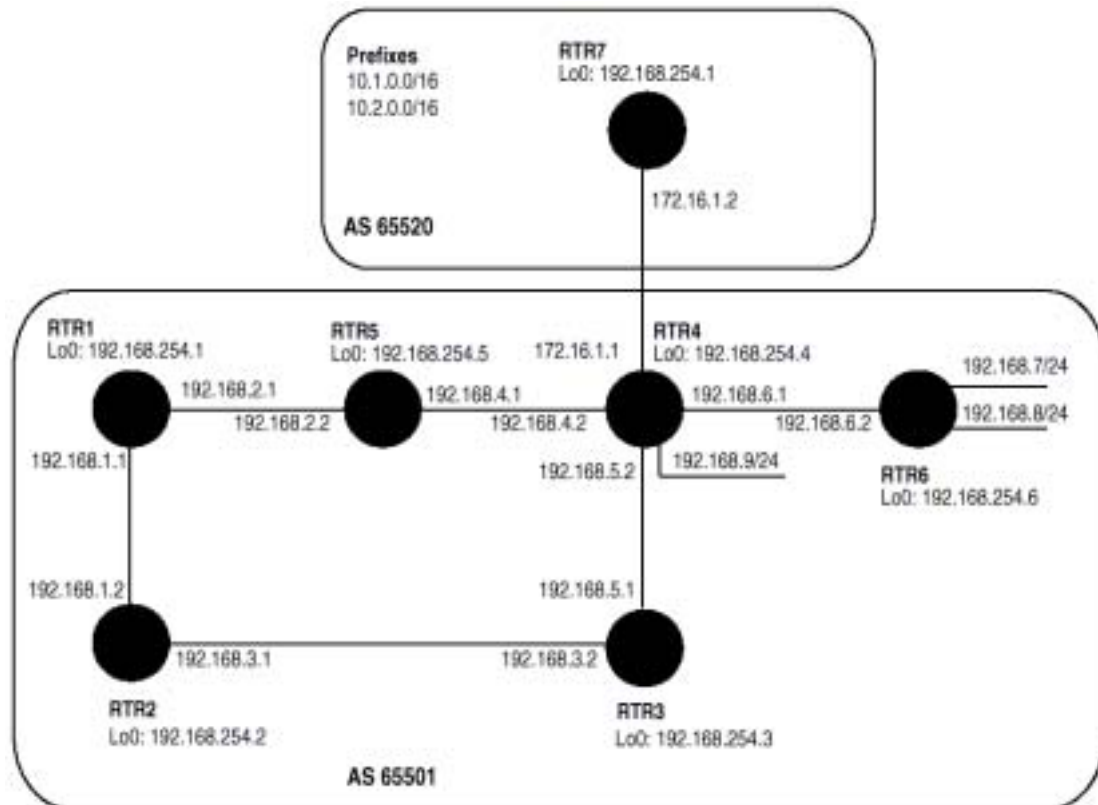


图1：演示网络的物理拓扑和逻辑拓扑

AS 65501中使用的IGP是OSPF，所有内部接口都位于区域0中。RTR4上连接AS65520的外部接口不运行OSPF。图2说明了逻辑BGP拓扑。在AS 65501内部配置了全网状IBGP，连接了除RTR6以外的所有路由器。该路由器上没有运行任何BGP。一条EBGP连接把AS 65501中的RTR4与AS 65520中的RTR7连接起来。EBGP连接的端点是两台路由器的外部物理接口。

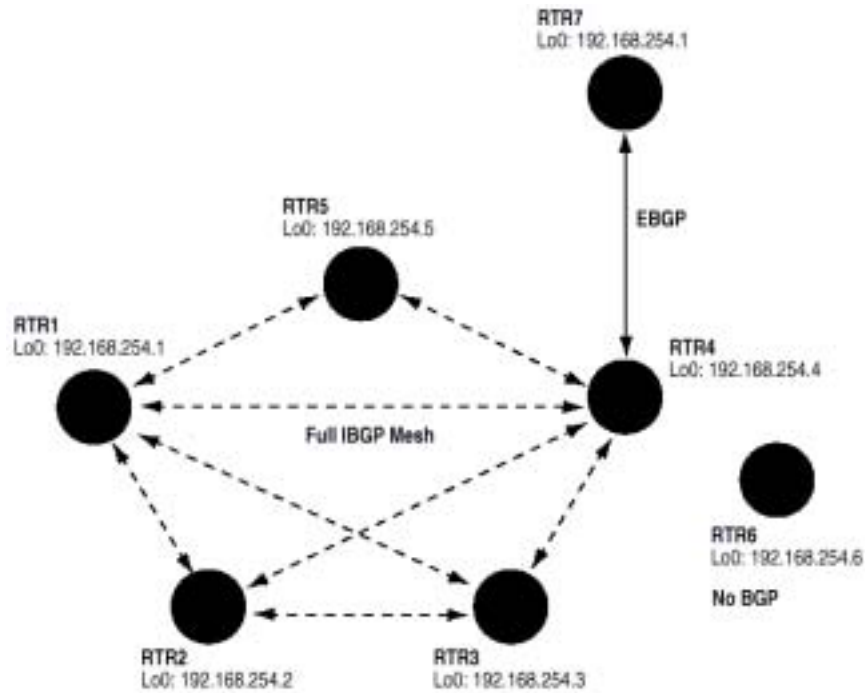


图2：演示网络的BGP拓扑

从RTR1到RTR4配置一条LSP，称为test_path，如图3所示。LSP明示路由对象(ERO)被指定为使用一个严格站，直到RTR2，这样LSP采用了一条与RTR1-RTR5-RTR4的OSPF最短路径不同的一条路径。LSP使用RSVP进行信令处理，但没有运行任何CSPF。

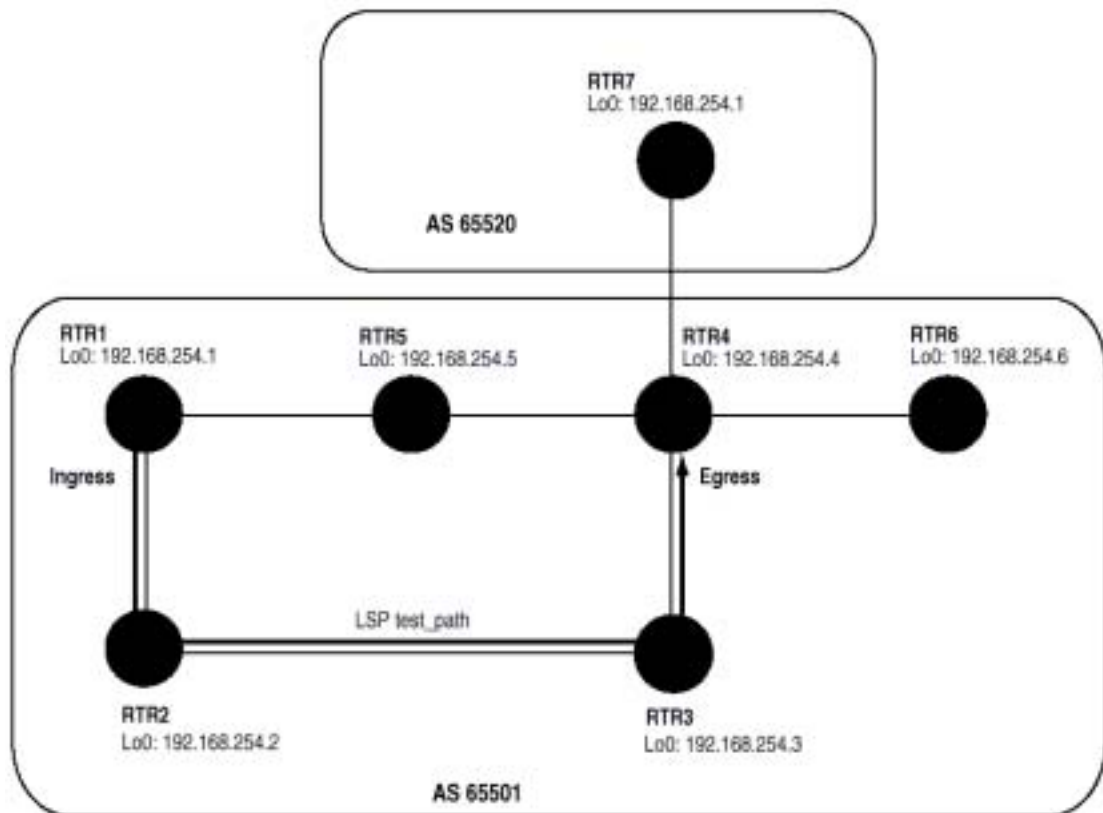


图3：演示网络的LSP拓扑

如需这七台路由器的基本路由选项、策略和协议配置，请参阅附录A。

路由信息数据库

JUNOS软件维护多个路由表，可以从这些路由表中提取最佳路径信息，并增加到转发表中。路由表的维护和使用与协议相关。与MPLS流量工程路由解析有关的路由表是inet.0、inet.3和mpls.0。

表1：MPLS流量工程路由解析使用的路由表

路由表	类型	说明
inet.0	单点广播路由表	所有单点广播协议将其路由信息输入到这个表中。
inet.3	MPLS路由表	RSVP把通过LSP可达的信宿输入到这个表中；这些信宿前缀一直都是LSP出口点的IP地址。
mpls.0	MPLS交换表	MPLS把标记绑定输入到这个表中，并使用信息把进入的MPLS标记分组交换到输出接口上。

根据附录A运行中列明的默认配置，RTR1的三个路由表如下：

```

jeff@RTR1> show route
inet.0: 20 destinations, 20 routes (18 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.1.0/24    *[Direct/0] 17:34:14
                 > via fxp0.0
192.168.1.1/32  *[Local/0] 17:34:14
                 Local
192.168.2.0/24  *[Direct/0] 17:34:14
                 > via fxp1.0
192.168.2.1/32  *[Local/0] 17:34:14
                 Local
192.168.3.0/24  *[OSPF/10] 17:33:27, metric 2
                 > to 192.168.1.2 via fxp0.0
192.168.4.0/24  *[OSPF/10] 17:33:22, metric 2
                 > to 192.168.2.2 via fxp1.0
192.168.5.0/24  *[OSPF/10] 17:33:22, metric 3
                 to 192.168.1.2 via fxp0.0
                 > to 192.168.2.2 via fxp1.0
192.168.6.0/24  *[OSPF/10] 17:33:22, metric 3
                 > to 192.168.2.2 via fxp1.0
192.168.7.0/24  *[OSPF/10] 17:33:22, metric 13
                 > to 192.168.2.2 via fxp1.0
192.168.8.0/24  *[OSPF/10] 17:33:22, metric 13
                 > to 192.168.2.2 via fxp1.0
192.168.9.0/24  *[OSPF/10] 17:33:22, metric 12
                 > to 192.168.2.2 via fxp1.0
192.168.254.1/32 *[Direct/0] 17:34:14

```

```

> via lo0.0
192.168.254.2/32    *[OSPF/10] 17:33:27, metric 1
> to 192.168.1.2 via fxp0.0
192.168.254.3/32    *[OSPF/10] 17:33:27, metric 2
> to 192.168.1.2 via fxp0.0
192.168.254.4/32    *[OSPF/10] 17:33:22, metric 2
> to 192.168.2.2 via fxp1.0
192.168.254.5/32    *[OSPF/10] 17:33:22, metric 1
> to 192.168.2.2 via fxp1.0
192.168.254.6/32    *[OSPF/10] 17:33:22, metric 3
> to 192.168.2.2 via fxp1.0
224.0.0.5/32      *[OSPF/10] 17:34:15, metric 1

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.254.4/32    *[RSVP/7] 17:33:22, metric 2, metric2 0
> to 192.168.1.2 via fxp0.0, label-switched-path test_path

mpls.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0                *[MPLS/0] 17:34:14, metric 1
                Receive
1                *[MPLS/0] 17:34:14, metric 1
                Receive

```

显示的第一个表是inet.0，其中包含18个活动的前缀。本处的表格说明了除本地和直接连接的子网外，所有前缀都是由OSPF输入的。

显示的第二个表是inet.3，其中包含一个项目。宿前缀是192.168.254.4/32，这是LSP test_path的出口，可以通过该LSP到达。

最后一个表是mpls.0。这个表中包含标记0和标记1的项目，在协议启动时由MPLS自动输入。在接收时，这些标记分别代表一个LSP端点和一个路由器告警。在mpls.0中没有输入任何其它标记，因为RTR1不是任何LSP的转接路由器。把RTR1的mpls.0表与RTR2的路由表进行比较。

```
jeff@RTR2> show route table mpls.0
```

```

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                *[MPLS/0] 20:12:23, metric 1
                Receive

```

```

1          *[MPLS/0] 20:12:23, metric 1
           Receive
100004     *[RSVP/7] 18:07:14, metric 1
           > to 192.168.3.2 via fxp1.0, label-switched-path test_path
  
```

这个表说明，带有标记100004的进入MPLS分组交换到接口fxp1上，并遵循LSP test_path。通过显示路径中四台路由器中每台路由器上的RSVP会话，您可以查看LSP的完整路径，包括标记绑定。

```
jeff@RTR1> show RSVP session brief
```

```
Ingress RSVP: 1 sessions
```

To	From	State	Rt	Style	Labelin	Labelout	LSPname
192.168.254.4	192.168.254.1	Up	0	1 FF	-	100004	test_path

```
Total 1 displayed, Up 1, Down 0
```

```
Egress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

```
Transit RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

```
jeff@RTR2> show RSVP session brief
```

```
Ingress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

```
Egress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

```
Transit RSVP: 1 sessions
```

To	From	State	Rt	Style	Labelin	Labelout	LSPname
192.168.254.4	192.168.254.1	Up	1	1 FF	100004	100004	test_path

```
Total 1 displayed, Up 1, Down 0
```

```
jeff@RTR3> show RSVP session brief
```

```
Ingress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

```
Egress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

```
Transit RSVP: 1 sessions
```

To	From	State	Rt	Style	Labelin	Labelout	LSPname
192.168.254.4	192.168.254.1	Up	1	1 FF	100004	3	test_path

```
Total 1 displayed, Up 1, Down 0
```

```
jeff@RTR4> show RSVP session brief
```

```
Ingress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

```
Egress RSVP: 1 sessions
```

To	From	State	Rt	Style	Labelin	Labelout	LSPname

```

192.168.254.4  192.168.254.1  Up      0  1 FF      3      - test_path
Total 1 displayed, Up 1, Down 0
Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

```

在所有四台路由器上，显示了 LSP 的入口和出口 IP 地址。显示的路由是 RTR1 上的入口路径，LSP 上转发的分组被分配一个 100004 标记。在 RTR2 上，LSP 是转接点，带有标记 100004 的进入分组被分配一个输出标记 100004。尽管这两个标记在数字上完全相同，但仍会在 RTR2 上发生标记交换；标记只在相邻的 LSR 之间有效。RTR3 把进入标记 100004 换成输出标记 3。作为出口的 RTR4 弹出标记 3，通过查找标准 IP 最长匹配路由，来路由接收的分组。

使用 traffic-engineering bgp

在启动 MPLS 时，默认状态下是启动命令 traffic-engineering bgp。由于它是默认值，因此它是隐含的，没有显示在 JUNOS MPLS 配置进程中。traffic-engineering bgp 命令导致 BGP 在解析路由时同时查询表 inet.0 和表 inet.3，如图 4 所示。所有其它单点广播路由协议都只查询 inet.0。这种行为的意义在于它不是解析信宿地址，而是解析下站地址，如本节中的实例所示。

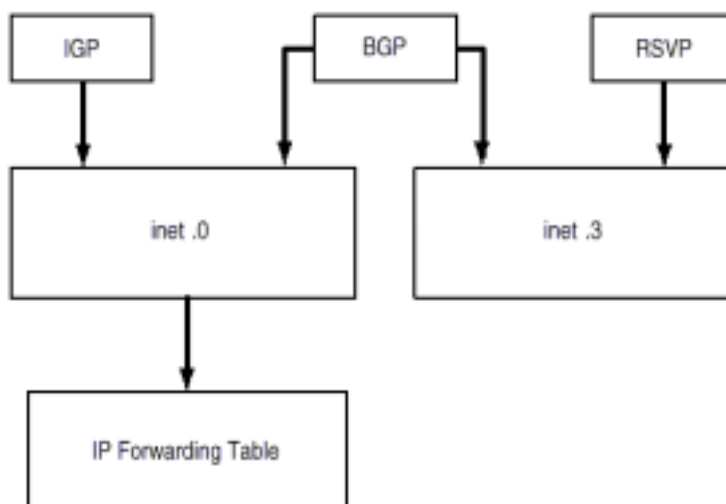


图 4：在默认状态下，BGP 同时使用 inet.0 和 inet.3

图 1 说明了 AS 65520 中有两个前缀通过 EBGP 广播到 AS 65501。这些前缀是 10.1.0.0/16 和 10.2.0.0/16。但是，重新检查上一节中显示的 RTR1 路由表后发现，这两个前缀都不在路由表中。注意，inet.0 的包头说明总共有 20 个已知前缀，其中 18 个是活动的，2 个是隐藏的。检查隐藏的前缀后，我们发现：

```

jeff@RTR1> show route table inet.0 hidden detail
inet.0: 20 destinations, 20 routes (18 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both
10.1.0.0/16 (1 entry, 0 announced)
      BGP      Preference: 170/-101

```

```
Next hop type: Unusable
State: <Hidden Int Ext>
Local AS: 65501 Peer AS: 65501
Age: 21 Metric2: 0
Task: BGP_65501.192.168.254.4+1037
AS path: 65520 I
BGP next hop: 172.16.1.2
Localpref: 100
Router ID: 192.168.254.4
```

10.2.0.0/16 (1 entry, 0 announced)

```
BGP Preference: 170/-101
Next hop type: Unusable
State: <Hidden Int Ext>
Local AS: 65501 Peer AS: 65501
Age: 21 Metric2: 0
Task: BGP_65501.192.168.254.4+1037
AS path: 65520 I
BGP next hop: 172.16.1.2
Localpref: 100
Router ID: 192.168.254.4
```

这两条路由通过IBGP从RTR4(192.168.254.4)上接收，但它们不能作为活动路由安装，因为不能使用路由器的下站地址。这些不可达的下站地址体现了默认的BGP行为，在IBGP会话中从外部学习的路由的下站属性保持不变。

在前缀来自AS 65520的情况下，下站地址是RTR7外部接口的地址172.16.1.2。当RTR1的BGP进程收到这些路由时，它会查找路由，以查看下站地址是否可达。迅速扫描上节中RTR1的路由表说明没有任何项目与172.16.1.2相匹配。因此，其宣告这些路由不可用。

为了缓和这种情况，使得路由可用，RTR4配置了一种策略，它覆盖外部路由的下站地址，把自己增加为下站。然后这种策略作为一种导出策略，应用于RTR4的内部对等。

```
bgp {
  group internal-peers {
    type internal;
    local-address 192.168.254.4;
    export next-hop-self;
    neighbor 192.168.254.1;
    neighbor 192.168.254.2;
    neighbor 192.168.254.3;
    neighbor 192.168.254.5;
  }
  group external-peer {
```

```

        type external;
        peer-as 65520;
        neighbor 172.16.1.2;
    }
policy-options {
    policy-statement next-hop-self {
        from protocol bgp;
        then {
            next-hop self;
        }
    }
}

```

在RTR1可以看到结果。

```

jeff@RTR1> show route receive-protocol bgp 192.168.254.4
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED    Lclpref AS path
10.1.0.0/16  192.168.254.4      100 65520 I
10.2.0.0/16  192.168.254.4      100 65520 I

```

下站地址变成RTR4的路由器ID 192.168.254.4。这时，流量工程BGP生效。BGP接收前缀，必须再次查询路由表，以学习怎样到达下站地址。当它查找一条到192.168.254.4的路由时，它发现下述项目：

```

jeff@RTR1> show route 192.168.254.4

inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.254.4/32 *[OSPF/10] 21:44:52, metric 2
                > to 192.168.2.2 via fxp1.0

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.254.4/32 *[RSVP/7] 21:44:52, metric 2, metric2 0
                > to 192.168.1.2 via fxp0.0, label-switched-path test_path

```

在inet.0和inet.3中都有一个192.168.254.4/32的项目。inet.0路由遵循OSPF最短路径，inet.3路由则是LSP test_path。注意，与OSPF路由相关的路由优先值是10，与RSVP路由相关的路由优先值则是7。由于首选较低的数字，因此选择LSP作为到达下站地址192.168.254.4更高优先的路由。

BGP现在解析下站的前缀，它把路由安装到路由表中的10.1.0.0/16和10.2.0.0/16，并以LSP作为到达该路由下站地址的路径。

```
jeff@RTR1> show route 10/8
```

```
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
10.1.0.0/16      *[BGP/170] 00:24:51, localpref 100, from 192.168.254.4
                  AS path: 65520 I
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
10.2.0.0/16      *[BGP/170] 00:24:51, localpref 100, from 192.168.254.4
                  AS path: 65520 I
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
```

这些结果表明了 MPLS 流量工程的主要应用，即为您提供一个工具，通过自治系统管理转发流量选取的路径。LSP 的构建方式是，其端点位于 BGP 边缘路由器上，其出口地址是边缘路由器的路由器 ID。然后边缘路由器把从 EBGP 对等中学习到的路由由下站地址转换成自己的路由器 ID。结果，可以使用到达这些路由器 ID 的 LSP 解析路由。

通过检查转发表，您可以进一步理解IGP和BGP与inet.0和inet.3的关系。看一下转发表中10.1.0.0/16的项目，下站地址显示为192.168.1.2。

```
jeff@RTR1> show route forwarding-table destination 10.1.0.0/16
```

```
Internet:
```

Destination	Type	RtRef	Nexthop	Type	Index	NhRef	Netif
10.1.0.0/16	user	0	192.168.1.2	Push	100004	fxp0.0	

192.168.1.2不是BGP路由的下站，而是LSP的转发下站；在本例中，是RTR2的接口(参见图1)。另一条线索是，路由类型说明了一种Push功能，Index表明了MPLS标记(标记被推送到入口LSR的分组上)。

192.168.254.4的转发项目如下：

```
jeff@RTR1> show route forwarding-table destination 192.168.254.4
```

```
Internet:
```

Destination	Type	RtRef	Nexthop	Type	Index	NhRef	Netif
192.168.254.4/32	user	1	192.168.2.2	ucst	25	12	fxp1.0

注意，这个项目的转发下站是192.168.2.2，这是RTR5的接口。这条路径是OSPF路由。这两个项目的含义是，尽管BGP已经决定到达192.168.254.4的最佳路径是通过LSP test_path，但该路由器仍通过OSPF路由转发信宿地址为192.168.254.4的分组。

这两条轨迹路由进一步说明了这一点。

```

jeff@RTR1> traceroute 10.1.1.1
traceroute to 10.1.1.1 (10.1.1.1), 30 hops max, 40 byte packets
 1  192.168.1.2 (192.168.1.2) 0.426 ms 0.237 ms 0.180 ms
    MPLS Label=100004 CoS=0 TTL=1 S=1
 2  192.168.3.2 (192.168.3.2) 0.347 ms 0.280 ms 0.270 ms
    MPLS Label=100004 CoS=0 TTL=1 S=1
 3  192.168.5.2 (192.168.5.2) 0.371 ms 0.295 ms 0.295 ms
 4  * * *
^C

```

```

jeff@RTR1> traceroute 192.168.254.4
traceroute to 192.168.254.4 (192.168.254.4), 30 hops max, 40 byte packets
 1  192.168.2.2 (192.168.2.2) 0.370 ms 0.197 ms 0.164 ms
 2  192.168.254.4 (192.168.254.4) 0.339 ms 0.270 ms 0.267 ms

```

第一条轨迹是到达属于10.1.0.0/16前缀的信宿，并遵循LSP。第二条轨迹到达192.168.254.4，并遵循OSPF路由。这些结果和我们在转发表中看到的相符。

本例中看到的行是图4中说明的关系的直接结果。转发表只基于inet.0中的路由。BGP可以查看inet.3，选择一条LSP，作为到达BGP前缀下站的最佳路径，并可以利用该LSP在inet.0增加一条路由。然后从inet.0路由中，在转发表内制作一个项目。在默认状态下，没有任何其它协议可以查询inet.3，inet.3路由没有输入到inet.0中。因此在inet.0中，只从到达该信宿的路由（OSPF路由）中为192.168.254.4创建转发项目。

使用IGP捷径

并不是所有人都使用next-hop self方法来解决EBGP下站问题。有些人首选在外部接口上以无源模式运行IGP。然后IGP识别外部子网，把路由输入到inet.0路由表的子网中。在解析AS外部路由的下站地址时，BGP将使用IGP路由。

在这种情况下不能使用Traffic-engineering BGP，因为BGP路由的下站地址不是任何出口LSR的路由器ID。因此，流量只映射到IGP路由，而不是LSP。

IGP捷径也称为traffic-engineering shortcuts，它提供了一个工具，AS中的链路状态IGP（OSPF或IS-IS）可以通过这个工具，在SPF计算中考虑LSP。如果使用无源外部接口，LSP则把LSP视为朝向LSP出口之上的信宿的单一数据链路。

在本例中，我们改变了RTR4的配置，以消除next-hop self策略，RTR7的外部接口fxp1则运行无源OSPF。

```

bgp {
  group internal-peers {
    type internal;
    local-address 192.168.254.4;
  }
}

```

```

neighbor 192.168.254.1;
neighbor 192.168.254.2;
neighbor 192.168.254.3;
neighbor 192.168.254.5;
}
group external-peer {
  type external;
  peer-as 65520;
  neighbor 172.16.1.2;
}
}
ospf {
  area 0.0.0.0 {
    interface fxp0.0;
    interface fxp2.0;
    interface fxp3.0;
    interface fxp4.0;
    interface fxp1.0 {
      passive;
    }
  }
}
}

```

RTR4现在把172.16.1.0/30子网广播到OSPF域中。

```
jeff@RTR1> show route 172.16.1/16
```

```
inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
172.16.1.0/30    *[OSPF/10] 00:03:46, metric 3
                 > to 192.168.2.2 via fxp1.0
```

RTR4把来自AS 65520的前缀广播到其IBGP对等中，而不必改变下站路由属性。当RTR1收到路由时，它再次查看inet.0和inet.3路由表，以查找到达BGP下站172.16.1.2的路由，它发现了OSPF路由。然后使用到达下站的OSPF路径，在inet.0中安装到AS 65520前缀的路由。

```
jeff@RTR1> show route 10/8 detail
```

```
inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
10.1.0.0/16 (1 entry, 1 announced)
```

```
*BGP Preference: 170/-101
```

```
Source: 192.168.254.4
```

```
Nexthop: 192.168.2.2 via fxp1.0, selected
```

```

State: <Active Int Ext>
Local AS: 65501 Peer AS: 65501
Age: 31 Metric2: 3
Task: BGP_65501.192.168.254.4+1037
Announcement bits (2): 2-KRT 4-BGP_Sync_Any
AS path: 65520 I
BGP next hop: 172.16.1.2
Localpref: 100
Router ID: 192.168.254.4
10.2.0.0/16 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Source: 192.168.254.4
    Nexthop: 192.168.2.2 via fxp1.0, selected
    State: <Active Int Ext>
    Local AS: 65501 Peer AS: 65501
    Age: 31 Metric2: 3
    Task: BGP_65501.192.168.254.4+1037
    Announcement bits (2): 2-KRT 4-BGP_Sync_Any
    AS path: 65520 I
    BGP next hop: 172.16.1.2
    Localpref: 100
    Router ID: 192.168.254.4

```

注意172.16.1.2的BGP下站和192.168.2.2的转发下站，其遵循OSPF最短路径。

参阅图 3，可以看到如果允许 OSPF 在 SPF 计算中包括 LSP，而且如果把 LSP 看作一条数据链路，那么 OSPF 会选择 LSP 作为到达 RTR4 下行子网的最短路径。

您只需在入口路由器上启动IGP捷径，因为这是执行SPF计算的路由器。RTR1的OSPF配置如下：

```

ospf {
  traffic-engineering shortcuts;
  area 0.0.0.0 {
    interface all;
  }
}

```

了解IGP捷径怎样影响图4中所示的协议和路由表关系，这一点非常重要。IGP对LSP出口点的下行子网执行SPF计算，但这些计算结果只输入inet.3表中(图5)。同时，IGP执行传统的SPF计算，把这些计算结果输入inet.0表中。其结果是，尽管IGP在inet.3表中生成了项目，但BGP仍是唯一可以查看该表格、以解析路由的协议。因此，转发到AS内部信宿仍使用inet.0 IGP路由，LSP则仅用于BGP下站解析。

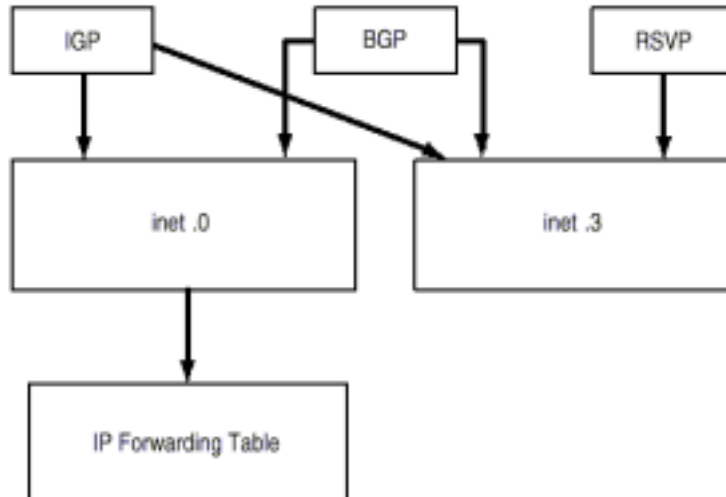


图5：IGP捷径允许IGP在inet.3中安装前缀

在启动IGP捷径后，RTR1的路由表如下：

```
jeff@RTR1> show route
```

```
inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```

10.1.0.0/16      *[BGP/170] 00:57:12, localpref 100, from 192.168.254.4
                  AS path: 65520 I
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
10.2.0.0/16      *[BGP/170] 00:57:12, localpref 100, from 192.168.254.4
                  AS path: 65520 I
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
172.16.1.0/30    *[OSPF/10] 00:51:39, metric 3
                  > to 192.168.2.2 via fxp1.0
192.168.1.0/24   *[Direct/0] 1d 15:19:11
                  > via fxp0.0
192.168.1.1/32   *[Local/0] 1d 15:19:11
                  Local
192.168.2.0/24   *[Direct/0] 1d 15:19:11
                  > via fxp1.0
192.168.2.1/32   *[Local/0] 1d 15:19:11
                  Local
192.168.3.0/24   *[OSPF/10] 00:51:39, metric 2
                  > to 192.168.1.2 via fxp0.0
192.168.4.0/24   *[OSPF/10] 00:51:39, metric 2
                  > to 192.168.2.2 via fxp1.0
192.168.5.0/24   *[OSPF/10] 00:51:39, metric 3
                  to 192.168.1.2 via fxp0.0
  
```

```

> to 192.168.2.2 via fxp1.0
192.168.6.0/24  *[OSPF/10] 00:51:39, metric 3
> to 192.168.2.2 via fxp1.0
192.168.7.0/24  *[OSPF/10] 00:51:39, metric 13
> to 192.168.2.2 via fxp1.0
192.168.8.0/24  *[OSPF/10] 00:51:39, metric 13
> to 192.168.2.2 via fxp1.0
192.168.9.0/24  *[OSPF/10] 00:51:39, metric 12
> to 192.168.2.2 via fxp1.0
192.168.254.1/32  *[Direct/0] 1d 15:19:11
> via lo0.0
192.168.254.2/32  *[OSPF/10] 00:51:39, metric 1
> to 192.168.1.2 via fxp0.0
192.168.254.3/32  *[OSPF/10] 00:51:39, metric 2
> to 192.168.1.2 via fxp0.0
192.168.254.4/32  *[OSPF/10] 00:51:39, metric 2
> to 192.168.2.2 via fxp1.0
192.168.254.5/32  *[OSPF/10] 00:51:39, metric 1
> to 192.168.2.2 via fxp1.0
192.168.254.6/32  *[OSPF/10] 00:51:39, metric 3
> to 192.168.2.2 via fxp1.0
224.0.0.5/32     *[OSPF/10] 1d 15:19:12, metric 1

```

inet.3: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

172.16.1.0/30    *[OSPF/10] 00:51:39, metric 3
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.5.0/24  *[OSPF/10] 00:51:39, metric 3
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.6.0/24  *[OSPF/10] 00:01:35, metric 3
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.7.0/24  *[OSPF/10] 00:51:39, metric 13
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.8.0/24  *[OSPF/10] 00:51:39, metric 13
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.9.0/24  *[OSPF/10] 00:51:39, metric 12
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.254.4/32  *[RSVP/7] 1d 15:18:19, metric 2, metric2 0
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
[OSPF/10] 00:51:39, metric 2
> to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.254.6/32  *[OSPF/10] 00:51:39, metric 3
> to 192.168.1.2 via fxp0.0, label-switched-path test_path

```

mpls.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```
0          *[MPLS/0] 1d 15:19:11, metric 1
          Receive
1          *[MPLS/0] 1d 15:19:11, metric 1
          Receive
```

在表inet.3中，RSVP和OSPF都输入路由。把OSPF路由与图1中所示的地址进行对比，您可以看出路由是到OSPF域内子网的路由，但它是LSP test_path出口的下行方向。这些路由是RTR4直接连接的子网，以及下行路由器RTR6直接连接的子网。

注意列出了192.168.5.0/24，但没有列出192.168.4.0/24。192.168.4.0/24是正常OSPF最短路径的一部分，因此不能看作RTR4的下行方向。LSP穿过192.168.5.0/24，但OSPF只查看LSP，因此把该子网看作RTR4的下行方向。

Inet.3中还包括172.16.1.0/30，在inet.0中可以看出，对两个AS 65520前缀的路由，BGP选择到该子网的LSP作为到达下站地址172.16.1.2的路径。在讨论traffic-engineering BGP的上一节中，BGP首选优先值为7的RSVP路径，而不是优先值为10的OSPF路径。但是，在本例中，到达表inet.0中172.16.1.0/30的OSPF路由，和到达表inet.3中同一前缀的OSPF路由，其优先值都是10，其量度都是3。这说明，当BGP在inet.0和inet.3中发现到下站地址的路由，而且两条路由的所有其它参数都相同时，那么将首选inet.3路由。

与IGP 捷径有关的最后运行说明涉及到 LSP 出口的地址。链路状态 SPF 树的端点是节点，用 OSPF 路由器 ID 或 IS-IS 系统 ID 表示。因此，只有 LSP 的出口地址是环回接口（在这个接口上配置 OSPF 路由器 ID 和 IS-IS 系统 ID）时，IGP 才能在 SPF 计算中使用 LSP。IGP 捷径不使用其端点是物理接口的 LSP，因为这些地址不代表节点。

下面详细说明了在启动IGP捷径后的两条BGP路由。把它与前面详细显示的没有IGP捷径的相同路由进行比较。

```
10.1.0.0/16 (1 entry, 1 announced)
  *BGP   Preference: 170/-101
         Source: 192.168.254.4
         Nexthop: 192.168.1.2 via fxp0.0, selected
         label-switched-path test_path
         State: <Active Int Ext>
         Local AS: 65501 Peer AS: 65501
         Age: 1:45:17 Metric2: 3
         Task: BGP_65501.192.168.254.4+1037
         Announcement bits (2): 2-KRT 4-BGP_Sync_Any
         AS path: 65520 I
         BGP next hop: 172.16.1.2
```

Localpref: 100
Router ID: 192.168.254.4

```

10.2.0.0/16 (1 entry, 1 announced)
  *BGP Preference: 170/-101
      Source: 192.168.254.4
      Nexthop: 192.168.1.2 via fxp0.0, selected
      label-switched-path test_path
      State: <Active Int Ext>
      Local AS: 65501 Peer AS: 65501
      Age: 1:45:17 Metric2: 3
      Task: BGP_65501.192.168.254.4+1037
      Announcement bits (2): 2-KRT 4-BGP_Sync_Any
      AS path: 65520 I
      BGP next hop: 172.16.1.2
      Localpref: 100
      Router ID: 192.168.254.4
  
```

在使用LSP到达一条外部链路时，IGP捷径的真正用处并不明显；但在使用一条LSP到达多台路由器上的一组外部链路时，IGP捷径的真正用处则变得非常明显。例如，您可能在多个城市中设有POP（业务点）。在每个POP，可能会有多台对等路由器。通过traffic engineering BGP，在所有POP中的所有对等路由器之间可能要求全网状LSP。随着对等路由器的数量增长，LSP的数量会呈指数级增长。LSP的管理开始遇到流量工程着力解决的同样的复杂度问题。但是，通过IGP捷径，可以为POP内部的一台路由器创建多条LSP，甚至为位于一个POP区域中央的一台路由器创建多条LSP。这样就降低了核心中的LSP总数，流量工程仍非常简单。

使用traffic-engineering bgp-igp

前面提到的traffic-engineering bgp 和 IGP捷径都仅用于解析BGP路由。但是，发送到AS内部宿的流量也可以映射到LSP上。在启动traffic-engineering bgp-igp时，RSVP把MPLS前缀安装到表inet.0中，而不是安装到表inet.3 (图6)中。结果，MPLS LSP现在被安装到下表中。

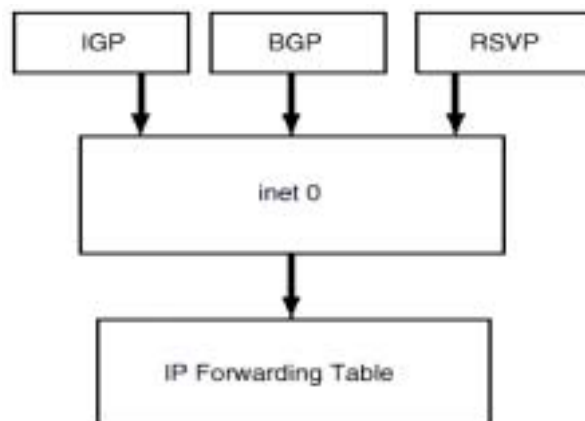


图6：traffic-engineering bgp-igp命令导致RSVP把前缀安装到inet.0中

traffic-engineering bgp-igp命令在JUNOS配置的MPLS部分中进行配置。在第一个实例中，RTR1被配置成运行traffic-engineering bgp-igp，而不是运行IGP捷径。

```
protocols {
  rsvp {
    interface all;
  }
  mpls {
    traffic-engineering bgp-igp;
    label-switched-path test_path {
      to 192.168.254.4;
      primary through_RTR2;
      no-cspf;
    }
    path through_RTR2 {
      192.168.254.2 strict;
    }
    interface all;
  }
  bgp {
    group internal-peers {
      type internal;
      local-address 192.168.254.1;
      neighbor 192.168.254.2;
      neighbor 192.168.254.3;
      neighbor 192.168.254.4;
      neighbor 192.168.254.5;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
    }
  }
}
```

其路由表和RTR1转发表结果如下：

```
jeff@RTR1> show route
```

```
inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```

10.1.0.0/16      *[BGP/170] 00:02:01, localpref 100, from 192.168.254.4
                 AS path: 65520 I
                 > to 192.168.2.2 via fxp1.0
10.2.0.0/16      *[BGP/170] 00:02:01, localpref 100, from 192.168.254.4
                 AS path: 65520 I
                 > to 192.168.2.2 via fxp1.0
172.16.1.0/30    *[OSPF/10] 00:20:26, metric 3
                 > to 192.168.2.2 via fxp1.0
192.168.1.0/24   *[Direct/0] 1d 23:57:14
                 > via fxp0.0
192.168.1.1/32   *[Local/0] 1d 23:57:14
                 Local
192.168.2.0/24   *[Direct/0] 1d 23:57:14
                 > via fxp1.0
192.168.2.1/32   *[Local/0] 1d 23:57:14
                 Local
192.168.3.0/24   *[OSPF/10] 00:20:26, metric 2
                 > to 192.168.1.2 via fxp0.0
192.168.4.0/24   *[OSPF/10] 00:20:26, metric 2
                 > to 192.168.2.2 via fxp1.0
192.168.5.0/24   *[OSPF/10] 00:20:26, metric 3
                 to 192.168.1.2 via fxp0.0
                 > to 192.168.2.2 via fxp1.0
192.168.6.0/24   *[OSPF/10] 00:20:26, metric 3
                 > to 192.168.2.2 via fxp1.0
192.168.7.0/24   *[OSPF/10] 00:20:26, metric 13
                 > to 192.168.2.2 via fxp1.0
192.168.8.0/24   *[OSPF/10] 00:20:26, metric 13
                 > to 192.168.2.2 via fxp1.0
192.168.9.0/24   *[OSPF/10] 00:20:26, metric 12
                 > to 192.168.2.2 via fxp1.0
192.168.254.1/32  *[Direct/0] 1d 23:57:14
                 > via lo0.0
192.168.254.2/32  *[OSPF/10] 00:20:26, metric 1
                 > to 192.168.1.2 via fxp0.0
192.168.254.3/32  *[OSPF/10] 00:20:26, metric 2
                 > to 192.168.1.2 via fxp0.0
192.168.254.4/32  *[RSVP/7] 00:23:12, metric 2, metric2 0
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path
                 [OSPF/10] 00:20:26, metric 2
                 > to 192.168.2.2 via fxp1.0
192.168.254.5/32  *[OSPF/10] 00:20:26, metric 1
                 > to 192.168.2.2 via fxp1.0

```

```
192.168.254.6/32    *[OSPF/10] 00:20:26, metric 3
                  > to 192.168.2.2 via fxp1.0
224.0.0.5/32     *[OSPF/10] 1d 23:57:15, metric 1
```

mpls:0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```
0                *[MPLS/0] 1d 23:57:14, metric 1
                  Receive
1                *[MPLS/0] 1d 23:57:14, metric 1
                  Receive
```

jeff@RTR1> show route forwarding-table

Internet:

Destination	Type	RtRef	Nexthop	Type	Index	NhRef	Netif
default	perm	0		rjct	9	1	
10.1.0.0/16	user	0	192.168.2.2	ucst	25	13	fxp1.0
10.2.0.0/16	user	0	192.168.2.2	ucst	25	13	fxp1.0
172.16.1.0/30	user	0	192.168.2.2	ucst	25	13	fxp1.0
192.168.1.0/24	intf	0		rslv		14	1 fxp0.0
192.168.1.0/32	dest	0	192.168.1.0	recv	12	1	fxp0.0
192.168.1.1/32	intf	0	192.168.1.1	locl	13	2	
192.168.1.1/32	dest	0	192.168.1.1	locl	13	2	
192.168.1.2/32	dest	0	0:90:27:c2:8a:a3	ucst	24	7	fxp0.0
192.168.1.255/32	dest	0	192.168.1.255	bcst	11	1	fxp0.0
192.168.2.0/24	intf	0		rslv		18	1 fxp1.0
192.168.2.0/32	dest	0	192.168.2.0	recv	16	1	fxp1.0
192.168.2.1/32	intf	0	192.168.2.1	locl	17	2	
192.168.2.1/32	dest	0	192.168.2.1	locl	17	2	
192.168.2.2/32	dest	0	0:d0:b7:75:ff:31	ucst	25	13	fxp1.0
192.168.2.255/32	dest	0	192.168.2.255	bcst	15	1	fxp1.0
192.168.3.0/24	user	0	192.168.1.2	ucst	24	7	fxp0.0
192.168.4.0/24	user	0	192.168.2.2	ucst	25	13	fxp1.0
192.168.5.0/24	user	0	192.168.2.2	ucst	25	13	fxp1.0
192.168.6.0/24	user	0	192.168.2.2	ucst	25	13	fxp1.0
192.168.7.0/24	user	0	192.168.2.2	ucst	25	13	fxp1.0
192.168.8.0/24	user	0	192.168.2.2	ucst	25	13	fxp1.0
192.168.9.0/24	user	0	192.168.2.2	ucst	25	13	fxp1.0
192.168.254.1/32	intf	0	192.168.254.1	locl	21	1	
192.168.254.2/32	user	1	192.168.1.2	ucst	24	7	fxp0.0
192.168.254.3/32	user	1	192.168.1.2	ucst	24	7	fxp0.0
192.168.254.4/32	user	1	192.168.1.2	Push	100004	fxp0.0	
192.168.254.5/32	user	1	192.168.2.2	ucst	25	13	fxp1.0
192.168.254.6/32	user	0	192.168.2.2	ucst	25	13	fxp1.0

224.0.0.0/4	perm	1	mdsc	8	1
224.0.0.1/32	perm	0	224.0.0.1 mcst	4	3
224.0.0.5/32	user	1	224.0.0.5 mcst	4	3
255.255.255.255/32	perm	0	bcst	5	1

MPLS:

Interface.Label	Type	RtRef	Nexthop	Type	Index	NhRef	Netif
default	perm	0		dscd	1	1	
0	user	0		recv	3	2	
1	user	0		recv	3	2	

注意，到达 192.168.254.4 的 LSP 与到达同一信宿的 OSPF 路由一起安装在表 inet.0 中。由于 RSVP 的路由优先值要好于 OSPF，因此选择 LSP 作为到 192.168.254.4 的最佳路由，并在转发表中安装 LSP。注意没有表 inet.3。

到目前为止，还没有怎么利用 traffic-engineering bgp-igp。但是，当启动 IGP 捷径时，IGP 可以再次在计算中使用 IGP。由于没有表 inet.3，因此计算结果输入到表 inet.0 中。

在同时使用 traffic-engineering bgp-igp 和 IGP 捷径时，RTR1 的配置如下：

```

protocols {
  rsvp {
    interface all;
  }
  mpls {
    traffic-engineering bgp-igp;
    label-switched-path test_path {
      to 192.168.254.4;
      primary through_RTR2;
      no-cspf;
    }
    192.168.254.2 strict;
  }
  interface all;
}
bgp {
  group internal-peers {
    type internal;
    local-address 192.168.254.1;
    neighbor 192.168.254.2;
    neighbor 192.168.254.3;
    neighbor 192.168.254.4;
    neighbor 192.168.254.5;
  }
}

```

```

}
ospf {
  traffic-engineering shortcuts;
  area 0.0.0.0 {
    interface all;
  }
}
}

```

其路由表结果如下：

```
jeff@RTR1> show route
```

inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

10.1.0.0/16      *[BGP/170] 00:03:05, localpref 100, from 192.168.254.4
                 AS path: 65520 I
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path
10.2.0.0/16      *[BGP/170] 00:03:05, localpref 100, from 192.168.254.4
                 AS path: 65520 I
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path
172.16.1.0/30    *[OSPF/10] 00:00:04, metric 3
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.1.0/24   *[Direct/0] 2d 00:23:15
                 > via fxp0.0
192.168.1.1/32   *[Local/0] 2d 00:23:15
                 Local
192.168.2.0/24   *[Direct/0] 2d 00:23:15
                 > via fxp1.0
192.168.2.1/32   *[Local/0] 2d 00:23:15
                 Local
192.168.3.0/24   *[OSPF/10] 00:03:05, metric 2
                 > to 192.168.1.2 via fxp0.0
192.168.4.0/24   *[OSPF/10] 00:03:05, metric 2
                 > to 192.168.2.2 via fxp1.0
192.168.5.0/24   *[OSPF/10] 00:00:04, metric 3
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.6.0/24   *[OSPF/10] 00:00:04, metric 3
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.7.0/24   *[OSPF/10] 00:00:04, metric 13
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.8.0/24   *[OSPF/10] 00:00:04, metric 13
                 > to 192.168.1.2 via fxp0.0, label-switched-path test_path

```

```

192.168.9.0/24  *[OSPF/10] 00:00:04, metric 12
                > to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.254.1/32  *[Direct/0] 2d 00:23:15
                > via lo0.0
192.168.254.2/32  *[OSPF/10] 00:03:05, metric 1
                > to 192.168.1.2 via fxp0.0
192.168.254.3/32  *[OSPF/10] 00:03:05, metric 2
                > to 192.168.1.2 via fxp0.0
192.168.254.4/32  *[RSVP/7] 00:49:13, metric 2, metric2 0
                > to 192.168.1.2 via fxp0.0, label-switched-path test_path
                [OSPF/10] 00:00:04, metric 2
                > to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.254.5/32  *[OSPF/10] 00:03:05, metric 1
                > to 192.168.2.2 via fxp1.0
192.168.254.6/32  *[OSPF/10] 00:00:04, metric 3
                > to 192.168.1.2 via fxp0.0, label-switched-path test_path
224.0.0.5/32     *[OSPF/10] 2d 00:23:16, metric 1

```

mpls:0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

0                *[MPLS/0] 2d 00:23:15, metric 1
                Receive
1                *[MPLS/0] 2d 00:23:15, metric 1
                Receive

```

OSPF选择LSP作为到达LSP出口下行信宿的最短路径。此外，由于IGP使用LSP到达外部子网172.16.1.0/30，因此BGP也在其两条路由中使用LSP。如果在RTR4上使用next-hop self，那么BGP仍将选择LSP，而不是IGP路径。

在把繁忙的流量路由到AS内部特定信宿时（如服务器场），这种方法特别实用。

与IGP捷径有关的重要一点是，不管是单独使用还是与traffic-engineering BGP-IGP一起使用，在LSP中都不会形成IGP邻居。IGP把LSP看作一条数据链路，但不会把出口路由器视作潜在的对等，也不在LSP上转发Hello消息。此外，它从不会在LSP上转发RSVP报文，以防止因疏忽在另一个LSP内部建立一条LSP。

安装单一前缀

与IGP捷径有关的一个可能问题是，它不能对路由表中安装的内容进行精细控制。有时只应把特定的信宿映射到一条LSP上。JUNOS软件提供了选项，允许把前缀逐个映射到LSP上。

在本例中，把RTR1和RTR4返回到基本配置。不管是traffic-engineering bgp-igp还是IGP捷径都不在RTR1上运行；RTR4没有在外接口上运行OSPF，也没有使用next-hop self。结果，AS 65520中的前缀10.1.0.0/24和10.2.0.0/24在RTR1上是不可达的。

```

jeff@RTR1> show bgp next-hop-database
10.1.0.0/16
    Source: 192.168.254.4 Next hop not resolved
10.2.0.0/16
    Source: 192.168.254.4 Next hop not resolved

```

为了使前缀可达，并保证 BGP 能够使用 LSP test_path 到达下站，可以以手动方式把前缀 172.16.1.0/30 与 LSP 关联起来，并安装在 RTR1 的 inet.3 中。

```

mpls {
  label-switched-path test_path {
    to 192.168.254.4;
    primary through_RTR2;
    install 172.16.1.0/30;
    no-cspf;
  }
  path through_RTR2 {
    192.168.254.2 strict;
  }
  interface all;
}

```

在inet.3中看到的结果是：

```

jeff@RTR1> show route table inet.3

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.1.0/30      *[RSVP/7] 00:03:10, metric 2, metric2 0
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
192.168.254.4/32  *[RSVP/7] 00:03:10, metric 2, metric2 0
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path

```

172.16.1.0/30只安装在inet.3中。BGP现在可以查表，找到下站地址172.16.1.2的匹配项目，然后安装到inet.0中外部前缀的路由上。

```

jeff@RTR1> show bgp next-hop-database
10.1.0.0/16
    Source: 192.168.254.4 Nexthop: 172.16.1.0
    172.16.1.0/30 MED 2 Next hop 192.168.1.2 Push 100004,
10.2.0.0/16
    Source: 192.168.254.4 Nexthop: 172.16.1.0

```

172.16.1.0/30 MED 2 Next hop 192.168.1.2 Push 100004,

jeff@RTR1> show route 10/8

inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```
10.1.0.0/16      *[BGP/170] 00:07:58, localpref 100, from 192.168.254.4
                  AS path: 65520 I
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
10.2.0.0/16      *[BGP/170] 00:07:58, localpref 100, from 192.168.254.4
                  AS path: 65520 I
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
```

正如单一的前缀可以安装在inet.3中一样，映射到LSP的单一前缀可以安装到AS内部路由使用的inet.0中。通过在install语句中追加active关键词，可以实现这种映射。例如，假设您希望把图1中AS 65501内部的子网192.168.8.0/24映射到LSP上，则RTR1的配置变动如下：

```
mpls {
    label-switched-path test_path {
        to 192.168.254.4;
        primary through_RTR2;
        install 172.16.1.0/30;
        install 192.168.8.0/24 active;
        no-cspf;
    }
    path through_RTR2 {
        192.168.254.2 strict;
    }
    interface all;
}
```

前缀与OSPF路径一起安装在inet.0中。此外，由于RSVP的优先值要高于OSPF，因此选择LSP作为转发路径。

jeff@RTR1> show route 192.168.8.0

inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```
192.168.8.0/24    *[RSVP/7] 00:01:46, metric 2, metric2 0
                  > to 192.168.1.2 via fxp0.0, label-switched-path test_path
                  [OSPF/10] 00:01:45, metric 13
                  > to 192.168.2.2 via fxp1.0
```

附录A

下面是演示网络中七台路由器的基本配置。在进行相关讨论时，我们已在本文正文中说明了这些配置的变动情况。

RTR1

```
routing-options {
    autonomous-system 65501;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path test_path {
            to 192.168.254.4;
            primary through_RTR2;
            no-cspf;
        }
        path through_RTR2 {
            192.168.254.2 strict;
        }
        interface all;
    }
    bgp {
        group internal-peers {
            type internal;
            local-address 192.168.254.1;
            neighbor 192.168.254.2;
            neighbor 192.168.254.3;
            neighbor 192.168.254.4;
            neighbor 192.168.254.5;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface all;
        }
    }
}
```

RTR2

```
routing-options {
  autonomous-system 65501;
}
protocols {
  rsvp {
    interface all;
  }
  mpls {
    interface all;
  }
  bgp {
    group internal-peers {
      type internal;
      local-address 192.168.254.2;
      neighbor 192.168.254.1;
      neighbor 192.168.254.3;
      neighbor 192.168.254.4;
      neighbor 192.168.254.5;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface all;
    }
  }
}
```

RTR3

```
routing-options {
  autonomous-system 65501;
}
protocols {
  rsvp {
    interface all;
  }
  mpls {
    interface all;
  }
  bgp {
    group internal-peer {
      type internal;
      local-address 192.168.254.3;
      neighbor 192.168.254.1;
      neighbor 192.168.254.2;
      neighbor 192.168.254.4;
      neighbor 192.168.254.5;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface all;
  }
}
```

RTR4

```
routing-options {
  autonomous-system 65501;
}
protocols {
  rsvp {
    interface fxp0.0;
    interface fxp4.0;
  }
  mpls {
    interface fxp0.0;
    interface fxp4.0;
  }
  bgp {
    group internal-peers {
      type internal;
      local-address 192.168.254.4;
      neighbor 192.168.254.1;
      neighbor 192.168.254.2;
      neighbor 192.168.254.3;
      neighbor 192.168.254.5;
    }
    group external-peer {
      type external;
      peer-as 65520;
      neighbor 172.16.1.2;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface fxp0.0;
      interface fxp2.0;
      interface fxp3.0;
      interface fxp4.0;
    }
  }
}
```

RTR5

```
protocols {
  rsvp {
    interface all;
  }
  mpls {
    interface all;
  }
  bgp {
    group internal-peers {
      type internal;
      local-address 192.168.254.5;
      neighbor 192.168.254.1;
      neighbor 192.168.254.2;
      neighbor 192.168.254.3;
      neighbor 192.168.254.4;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface all;
    }
  }
}
```

RTR6

```
protocols {
  ospf {
    area 0.0.0.0 {
      interface all;
    }
  }
}
```

RTR7

```
routing-options {
  static {
    route 10.1.0.0/32 {
      reject;
      install;
    }
    route 10.2.0.0/32 {
      reject;
      install;
    }
  }
  autonomous-system 65520;
}
protocols {
  bgp {
    group external-peers {
      type external;
      export statics;
      peer-as 65501;
      neighbor 172.16.1.1;
    }
  }
}
policy-options {
  policy-statement statics {
    from protocol static;
    then accept;
  }
}
```

缩略语

AS	自治系统
BGP	边界网关协议
CSPF	限制最短路径优先
EBGP	外部边界网关协议
ERO	明示路由对象
IGP	内部网关协议
IS-IS	中间系统到中间系统
LSP	标记交换路径
LSR	标记交换路由器
MPLS	多协议标记交换
OSPF	开放最短路径优先
POP	业务点
RSVP	资源预留协议
SPF	最短路径优先