

**OSPF Routing
With Cross-Address Family
MPLS Traffic Engineering Tunnels**

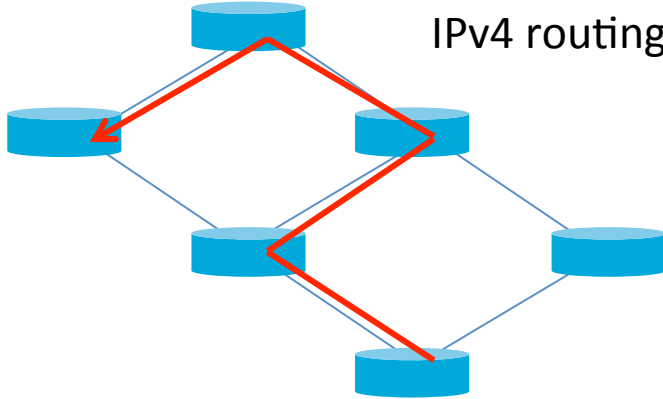
draft-smirnov-ospf-xaf-te-00

Introduction

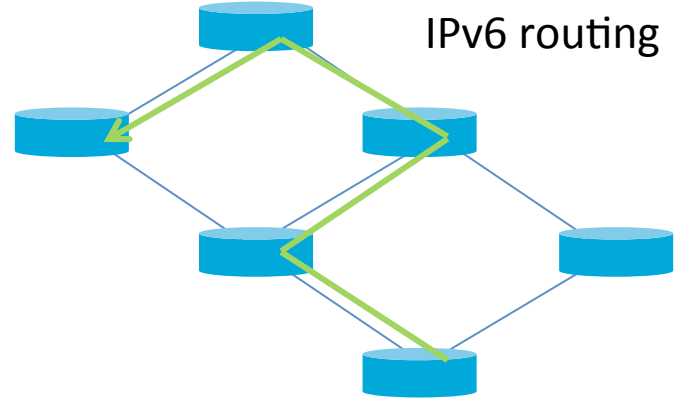
- RFC 3630 specified MPLS TE traffic engineering capabilities for OSPFv2
- RFC 5329 specified MPLS TE capabilities for OSPFv3
- If network is dual stack IPv4/IPv6 and runs both OSPFv2 and OSPFv3 then it is possible to enable MPLS TE in both:
 - OSPFv2 and OSPFv3 TE are “ships in the night”
 - TE infrastructure is duplicated: two MPLS TE tunnels along each path – one is signalled by OSPFv2 and used for IPv4 traffic and one is signalled by OSPFv3 and used for IPv6 traffic
- Duplication of TE may be undesirable:
 - Resource and scalability implications
 - Operational implications (operator may want to concentrate on service level delivered by TE tunnel and ignore details of which IP version is used by clients)

Introduction (cont.)

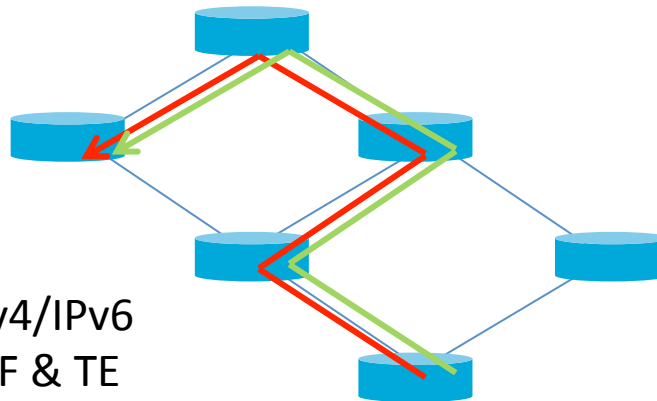
OSPFv2, RFC 3630
IPv4 TE signalling
IPv4 routing



OSPFv3, RFC 5329
IPv6 TE signalling
IPv6 routing



Modern dual-stack IPv4/IPv6
network running OSPF & TE

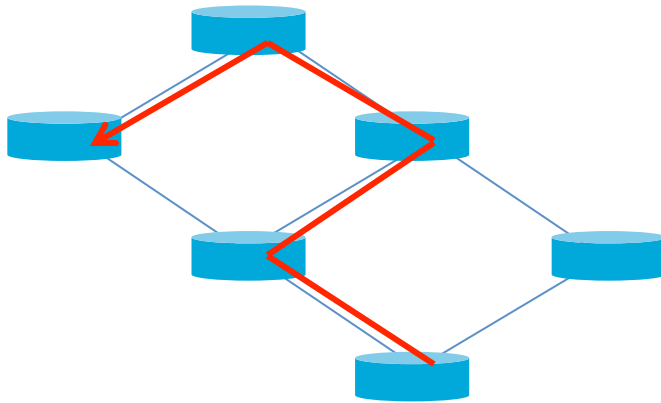


Problem definition

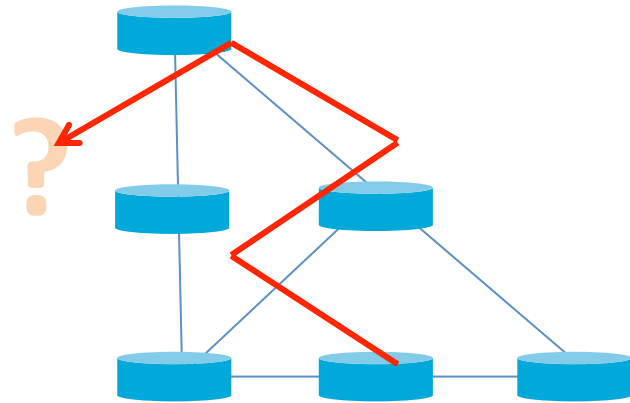
- Want to decouple TE tunnel signalling from the routing decision – cross-address family MPLS TE tunnels
 - For example, OSPFv2 is used to signal MPLS TE tunnels and to route IPv4 traffic; OSPFv3 routes IPv6 traffic and uses those MPLS TE tunnels for IPv6 routing
- Difficulty: to use TE shortcuts SPF must identify Rtr LSA of tunnel tail-end router (i.e. where the tunnel terminates)
- This knowledge comes for free when the same protocol instance is used to both propagate signalling information and to make routing decisions
- But when a protocol instance wants to make routing decisions for TE tunnels signalled by another protocol instance, identifying Rtr LSA of the tail-end router is problematic

Problem definition (cont.)

OSPFv2 database



OSPFv3 database



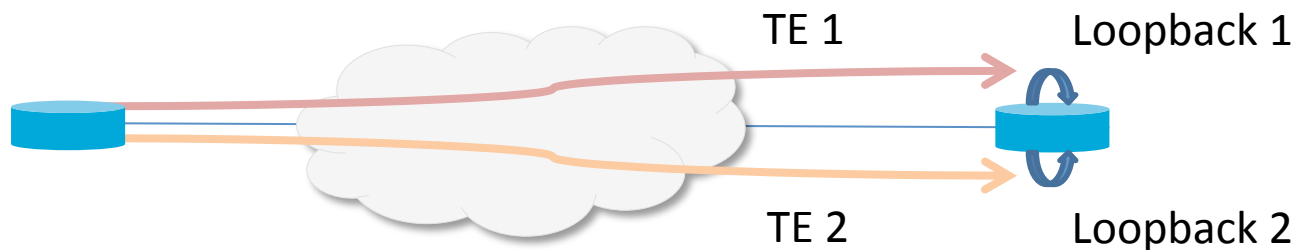
- OSPFv2 used to signal TE tunnel
- What is known about the tunnel:
 - Tunnel destination IPv4 address (the one advertised by OSPFv2 in Router Address TLV). Meaningless in context of IPv6/OSPFv3
 - OSPFv2 Router ID of tunnel's tail-end router. Has meaning in OSPFv2 LSDB
- There is no reliable way for OSPFv3 to determine where the tunnel terminates

Simple (but problematic) solution

- Simple (for developers) solution – require that OSPFv2 and OSPFv3 instances have the same Router ID
- In this case Rtr ID of router which advertised in OSPFv2 TE LSA with Router Address TLV can be used to lookup Rtr LSA in OSPFv3 LSDB
- Very simple to implement – not-so-simple for operators:
 - OSPFv2 and OSPFv3 are independent protocols, ensuring Rtr ID consistency is manual
 - Tunnel head-end has no built-in protocol means to verify that tail-end conforms to this “gentlemen’s agreement”
 - Problems with Router ID assignment are rare but DO routinely happen
- What will happen if tail-end router does not use the same Router ID for OSPFv2 and OSPFv3 instances:
 - Best case: OSPFv3 can’t find Rtr LSA with the same ID in its LSDB -> can’t use TE tunnels to this tail-end router
 - Worst case: Rtr ID exists but is assigned to another router -> incorrect routing decision -> suboptimal routing and risk of routing loops

Necessary digression – RFC 5786

- RFC 5786: “Advertising a Router's Local Addresses in OSPF Traffic Engineering (TE) Extensions”
- Defines TE (sub-)TLVs to advertise IP addresses local to the router but other than Router Address TLV
- Application: to advertise additional addresses to terminate TE tunnels



RFC 5786 – sub-TLV format

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          1          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Prefix Len 1 |          IPv4 Prefix 1          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Prefix 1 cont. |                                     :
+---+---+---+---+---+~                                     :
:                                     .                                     :
~                                     .               +---+---+---+---+---+
:                                     .               | Prefix Len n |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          IPv4 Prefix n          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

```

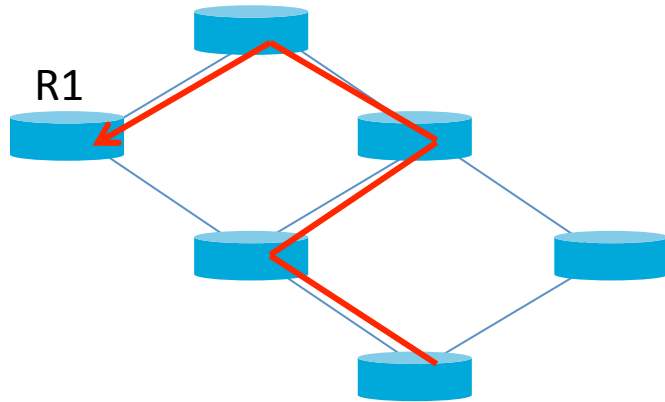
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          2          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Prefix Len 1 | Prefix 1 Opt. | IPv6 Prefix 1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 Prefix 1 cont. |                                     :
:                                     .                                     ~
~                                     .                                     :
:                                     .                                     :
:               +---+---+---+---+---+---+---+---+---+---+---+---+
:               | Prefix Len n | Prefix n Opt. |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          IPv6 Prefix n          |                                     :
|                                     :
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---
```


Proposed solution

- Use Node Local Address sub-TLV defined by RFC 5786 to help non-signalling OSPF instance identify tunnel tail-end routers
- Tunnel termination IP address is known but it is from address-family 'foreign' to the instance
- In non-signalling OSPF instance Node Local Address sub-TLVs advertise cross-address family IP address(es) where TE tunnels can be terminated
- Not all IP addresses are advertised; typically it is just one address – the one which signalling OSPF instance advertises in TE Router Address TLV
- Non-signalling OSPF instance then takes XAF tunnel termination address and looks up its OWN LSDB which router advertised TE LSA with matching Node Local Address

Proposed solution (example)

OSPFv2 database



Each router advertises:

RFC 3630 TE signalling

(optional) additional IPv4 RFC 5786

R1:

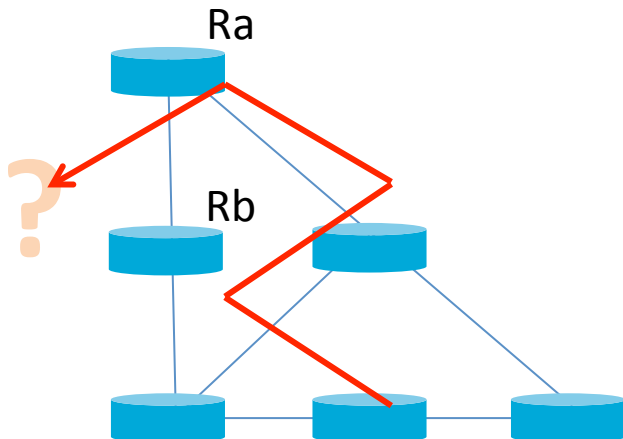
RID 192.168.1.1

Router Address TLV 1.1.1.1

TE tunnel:

Dest addr 1.1.1.1

OSPFv3 database



Each router advertises:

(optional) RFC 5329 TE signalling

(optional) additional IPv6 RFC 5786

XAF IPv4 Node Local Address sub-TLV

Ra:

RID 10.1.1.1

Router Address TLV AAA::1

XAF Local Addr 2.2.2.2

Rb:

RID 10.1.1.2

RA TLV AAA::2

XAF LA 1.1.1.1

Where does the TE tunnel terminate? On Rb, obviously

Discussion