Routing and MPLS/IP

Olof Hagsand KTH CSC
Literature

• Lecture slides and lecture notes (on web)
• Reference
  - JunOS Cookbook: Chapter 14
Background

• MPLS - Multiprotocol Label Switching
  • Originally thought to simplify IP forwarding
    - Small label lookup instead of longest prefix match
  • Roots in ATM (Asynchronous Transfer Mode)
    - Early 90s, most telecoms thought ATM would take over all data- and tele-communication
  • MPLS/IP was standardized in IETF in mid 90s.
  • GMPLS can be used for optical networking such as management of wavelengths: “lambdas”
  • MPLS/TP is currently being standardized
    - Transport Profile
    - MPLS/TP is independent of IP and for non-signalled low-level optical networks
MPLS Advantages

Originally, the motivation was speed and cost. But routers does IP lookup in hardware at very high speeds. Current advantages:

• Label switching can be used for *traffic engineering*
  - Aggregating a class of traffic
  - Guarantees: Allocation of resources
  - Constrained routing: load, bw, etc.

• Labels can be used to forward using other fields than destination address

• Label switching can be used to support VPNs – virtual private networks
Where is MPLS used?

• MPLS is used as a tunneling technique within an operator's internal IP network
  Tunneling characteristics - traffic is isolated
  VPNs
  Traffic engineering - control bandwidths and links
• MPLS is not used
  In traditional enterprise networks
  Between operators (inter-domain)
Why MPLS?

• MPLS gives a simple tunneling mechanism integrated with IP
• Another IP-based tunneling protocol could give the same service
  IP in IP
• But MPLS has a nice toolbox and is “easy” to configure
• Alternatives
  Pure IP networking: “manage tunnels yourself”
  Provider backbone bridging (IEEE 802.1ah)
MPLS Terminology

MPLS uses some new terminology (MPLS ~ IP)

• LSR: Label Switching Router ~ Router

• LER: Label Edge Router ~ Border router
  - Alternative: PE - Provider Edge / CE - Customer Edge
  - Also: Egress/Ingress/Transit LSR

• LSP: Label Switched Path ~ Tunnel

• FEC: Forwarding Equivalence Class ~ Flow

• Label distribution ~ Routing

• LFIB: Label Forwarding Information Base ~ FIB
Control and Data Planes

**LSR**
- Control Plane (Binding Layer)
- LFIB
- Data Plane (Forwarding Layer)
- MPLS packet

**Routing**
- Control Plane (Routing Layer)
- FIB
- Data Plane (Forwarding Layer)
- IP packet

**Routing**
- OSPF
- IS-IS
- BGP

**Label Distribution**
- LDP-CR
- RSVP-TE
- MP-BGP
Labels

• A label is an integer identifying a FEC (a flow).
• You cannot have globally or network-unique labels
  - Too complex to negotiate
  - Too large labels
• Labels are unique only between two nodes
• Labels 0-1048575.
  - 0-15 reserved by the IETF.
• Labels change at each node as a packet traverses a path
• You can set labels manually (worse than static routing), or use label distribution
• Example of Label Forwarding Information Base LFIB (MPLS forwarding table):

<table>
<thead>
<tr>
<th>In-Interface</th>
<th>in-label</th>
<th>op</th>
<th>out-label</th>
<th>out-interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>if 3</td>
<td>99</td>
<td>swap</td>
<td>203</td>
<td>if1</td>
</tr>
<tr>
<td>if 3</td>
<td>333</td>
<td>swap</td>
<td>978</td>
<td>if2</td>
</tr>
</tbody>
</table>
Encapsulation

- MPLS uses a 32-bit *shim* header:
  - Label: Value for table lookup in router
  - TC: Traffic class field, can be used as class-of-service for QoS
  - Stack: Indicates that the bottom of a stack of labels has been reached
  - TTL: Time To Live (resembles IP TTL)

- Shim headers may be concatenated into *stacks*
Forwarding Equivalence Class (FEC)

Sort packets into different classes – classify them.
Classification is a more general form of lookup
Example 1: All packets to one destination: ipdst = 192.168.20.33
Example 2: All UDP packets with the ToS field set to 0x42 from sub-network 192.168.20.0/24

Such a subset is called a Forwarding Equivalence Class (FEC)

MPLS binds labels to FECs – Labelling

FECs granularity
Coarse – good for scalability
Fine – good for flexibility

What defines FECs?
“Something else”: eg BGP routing table, VPN, packet filters, etc.
I.e., The meaning of a FEC (the semantics) is added by the overlying application
Label operations

- Push a label (typically at ingress)
  - Double push (label stacking)
- Swap a label
  - Made by internal LSRs / P routers
- Pop a label
  - Typically at egress or pen-ultimate LSR
- Label operations are interface-specific
  - Since labels are unique between LSRs
The border router (LER) classifies packets into FECs. It binds a label to the packet—actually, it maps the FEC to an LSP which in turn defines a label. It pushes an MPLS header on the packet and sends it on the outgoing interface of the LSP.
Label swapping

- A router (LSR) makes a label lookup and swaps the label
- Rewrites the MPLS header
- And sends it further on the LSP
Label popping

• The border router (LER) pops the MPLS packet
• And then forwards it as usual depending on the packets protocol
• Example: pkt is an IP packet --> pkt is sent to IP forwarding
  - Thus both MPLS and IP forwarding on same node!
Pen-ultimate popping

- To make it easier for the border router, pop the label on the previous router (pen-ultimate)
- The pen-ultimate LSR does MPLS pop
- The LER does only IP routing

<table>
<thead>
<tr>
<th>In-Interface</th>
<th>in-labelop</th>
<th>out-label</th>
<th>out-interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>if 3</td>
<td>203</td>
<td>pop</td>
<td>if1</td>
</tr>
</tbody>
</table>
Special labels operations

• 0: IPv4 explicit NULL
  - Downstream LSR should pop label unconditionally
  - Popped packet is an IPv4 datagram

• 1: Router alert
  - Deliver to control plane – do not forward

• 2: IPv6 explicit-NULL
  - Downstream LSR should pop unconditionally
  - Popped packet is an IPv6 datagram

• 3: Implicit-NULL
  - Pop immediately and treat as IPv4 packet
  - Note: This label does not actually appear on link (virtual)
  - Use for pen-ultimate popping!
MPLS Label Switched Paths

Example: IP

Eth hdr   Label hdr   IP hdr   IP payload

20-bit label

203 IP pkt

130 IP pkt

1345 IP pkt

IP pkt

Push label

Swap label

Pop label

Route packet

Pen-ultimate popping

3-bit exp

1-bit stack

8-bit TTL
Label Stacking

- Used inside a network for example in VPNs
- An *inner* label carrying edge information (e.g., which VPN) is transferred through the domain (33 and 55 in the example)
- *Outer* label is used for transfer through the network
- Push both labels at ingress.
- Pop outer label at pen-ultimate LSR
Typical use: MPLS for **transit**

*(lab scenario)*

- Use an IGP to compute internal routes
- Setup LSPs between border routers using the IGP
  - Eg border routers may set up a *full-mesh* of LSPs
- Send *transit traffic* via LSPs (src and dst outside the AS)
- But still send *internal* traffic via IP (src or dst inside the AS)
- External routes need not be distributed to non-border routers, so we do not need IBGP there
  - A *BGP-free core*
- Only the border routers need to speak BGP
  - This is considered an advantage
MPLS for transit

- Having MPLS in the core thus means that you do not need IBGP in internal routers
  - You still (always) need IBGP between border routers
- Thus, MPLS is an alternative to using IGP internally for external routes (bad choice) or IBGP (suffers from scaling)
- Note that interbal traffic still uses IP, only transit goes in MPLS
- We need some tricks in the border routers to make transit traffic follow LSPs and not IP
  - separation between transit and internal traffic
  - In JunOS this is done with inet.3 for next-hop routes (last slide)
Label Distribution

• Labels need to be assigned and LFIBs programmed
• A signaling protocol distributes labels
  – Creates an LSP through an MPLS network
• There are different ways to do this
  • 1. Make a new protocol
    – LDP – Label Distribution Protocol
  • 2. Extend existing protocols
    – BGP – Border Gateway Protocol
    – RSVP – Resource Reservation Protocol
• These protocols all distributes labels
  – But they are somewhat different and can be combined to transfer different labels, eg BGP+RSVP, where BGP transfers *inner* labels and RSVP negotiate *outer* labels.
  – RSVP is great for bandwidth reservation,...
Label distribution and IGP

• Label distribution protocols typically rely on an IGP (e.g., OSPF/IS-IS) to find the shortest path of their LSP.
  - Or use source-routing to do constrained SPF or fixed-path routing
• They then assign labels to the LSP on the path.
• Normally, labels are assigned in \textit{upstream} direction
• During setup of LSPs, traffic may be \textit{black-holed}
  - Discrepancy between IGP and label distribution
  - Same is true during reconvergence (IGP routes change)
Resource Reservation Protocol

• RSVP is a network control protocol used to express quality of service (QoS).
  - Binds a QoS request to a flow
• RSVP delivers QoS reservations along a path from source to destination(s).
  - No routing: IGP computes path
  - Uses "soft-state": paths are recomputed when routing changes
• RSVP-TE is used for traffic engineering for MPLS
Two types of messages to set up reservations:

- **PATH message**
  - From a sender to one or several receivers, carrying TSpec and classification information provided by sender

- **RESV message**
  - From receiver, carrying FlowSpec indicating QoS required by receiver
MPLS support in RSVP-TE

- RSVP-TE has been adapted to support MPLS
- New objects defined carrying labels
- In this way, RSVP-TE can express QoS (Tspec, FlowSpec) associated with an LSP
  - Network resources are then bound to that LSP throughout the network.
Traffic engineering with RSVP-TE/MPLS

- RSVP can reserve resources in the network
- RSVP can signal alternative paths using
  - Constrained shortest path (e.g., using allocated bandwidth)
  - Strict/loose source routing
- Why? Traffic distribution, alternate paths
  - Difficult to do with regular routing protocols.
- Example:
  - LSP strict source routing: A, B, C, D, E (20 Mbps)
- See routing algorithms lecture: CSPF
Protection switching in MPLS

• Assume a primary LSP is signalled from A-D via B and C
• If a link or node goes down, how is reliability ensured?
• There are several issues and techniques:
  - Detection of failure
  - IGP re-route
  - Path protection
  - Local protection

To think about
  - Switchover latency
  - Over-reservation
Detection of failure

- SDH/SONET does protection switching at ~50ms.
- Routing protocol Hello's are typically on 1s-10s.
- Physical level detection.
  But not all links support this.
- Instead, send packets very often
- MPLS pings along the LSP
  Send many packets and detect losses
- BFD - Bidirectional forwarding detection
  Send many packets and detect losses
  Generic technique for other protocols: OSPF, BGP, etc
- But sending many packets per LSP has its cost in bandwidth use
  And CPU usage if not done in hw
IGP reroute

• When the underlying IGP detects a failure, it will reroute around the failure, and thus RSVP-TE will send its PATH and RESV messages on the new route and the LSP will eventually establish itself using the new route.

• The cleanest solution but may be slow if IGP protection switching and RSVP failover is slow.

• This is done in the lab

![Diagram showing IGP reroute with nodes A, B, C, D, E, and F, and routes Initial LSP and Switchover LSP.]
Path protection

• Compute a secondary path in advance
• Switchover when the primary path fails using BFD/MPLS pings
• This is also done in the lab
Local protection (FRR)

- Protected LSP: ready-made detours
- The repair is made locally by pre-computed detour LSPs
  - Fast switchover since reroute made locally
- A detour is an extra LSP from a node in the path to a merge point
  - Link or node protection
  - One-to-any or "facility" LSP

- Example (link protection)
  - Protected LSP: A->B->C->D
  - Detours: A->E->B; B->E->F->C; C->F->D (only 2nd shown in fig)
Summary: MPLS in a transit network

Putting it all together:
• IGP is used to routes between all routers in the network
• MPLS is used to carry transit traffic between border routers (PE)
• RSVP-TE is used to signal MPLS LSPs between border routers.
• RSVP-TE is used to reserve bandwidth in LSPs
• RSVP-TE used to compute alternative paths for switchover
• BGP is used for all external routes.
• This is the lab scenario.
Lab: Using MPLS for transit: BGP-free core
MPLS in JunOS

• See
  http://www.juniper.net/techpubs/software/junos/junos94/swconfig-mpls-apps

• Example:

  Enable mpls on all forwarding interfaces
  Enable icmp in mpls for debugging (traceroute)
  Setup LSPs (using explicit path setup: no cspf)

```plaintext
interface so-0/0/0 {
    unit 0 {
        family mpls; # Enable mpls address family
    }
}
protocols mpls {
    icmp-tunneling; # Enable icmp for debugging
    interface so-0/0/0.0; # Include interface in mpls forwarding
    label-switched-path btoc { # Define an LSP
        to 193.10.255.6; # LSP end-point
        no-cspf; # Enable explicit-path computation
    }
    rsvp {
        interface so-0/0/0.0; # Enable rsvp on interface
    }
}
```

• cspf - Constrained Path Shortest Path
  Dont use it in lab - use explicit routing.
### MPLS show commands

> show mpls lsp

**Ingress LSP: 1 sessions**

<table>
<thead>
<tr>
<th>To</th>
<th>From</th>
<th>State</th>
<th>Rt</th>
<th>P</th>
<th>ActivePath</th>
<th>LSPname</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.10.255.6</td>
<td>193.10.255.5</td>
<td>Up</td>
<td>1</td>
<td>*</td>
<td></td>
<td>btoc</td>
</tr>
</tbody>
</table>

**Egress LSP: 1 sessions**

<table>
<thead>
<tr>
<th>To</th>
<th>From</th>
<th>State</th>
<th>Rt</th>
<th>Style</th>
<th>Labelin</th>
<th>Labelout</th>
<th>LSPname</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.10.255.5</td>
<td>193.10.255.6</td>
<td>Up</td>
<td>0</td>
<td>FF</td>
<td>3</td>
<td>-</td>
<td>ctob</td>
</tr>
</tbody>
</table>

**Transit LSP: 2 sessions**

<table>
<thead>
<tr>
<th>To</th>
<th>From</th>
<th>State</th>
<th>Rt</th>
<th>Style</th>
<th>Labelin</th>
<th>Labelout</th>
<th>LSPname</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.10.255.5</td>
<td>193.10.255.6</td>
<td>Up</td>
<td>1</td>
<td>FF</td>
<td>299792</td>
<td>3</td>
<td>ctob</td>
</tr>
<tr>
<td>193.10.255.6</td>
<td>193.10.255.5</td>
<td>Up</td>
<td>1</td>
<td>FF</td>
<td>299776</td>
<td>3</td>
<td>btoc</td>
</tr>
</tbody>
</table>

> show route protocol rsvp

299776  *[RSVP/7] 3d 18:23:44, metric 1
  > via so-0/1/0.0, label-switched-path btoc

299776(S=0)  *[RSVP/7] 3d 18:23:44, metric 1
  > via so-0/1/0.0, label-switched-path btoc

299792  *[RSVP/7] 3d 18:23:37, metric 1
  > via so-0/0/0.0, label-switched-path ctob

299792(S=0)  *[RSVP/7] 3d 18:23:37, metric 1
  > via so-0/0/0.0, label-switched-path ctob

> show mpls interface

> show rsvp interface

> show rsvp neighbour

> ...
EBGP nexthop: recursive lookup

RTC:s routing table alternatives:
Next-hop self is necessary for BGP to use MPLS!

<table>
<thead>
<tr>
<th>Route</th>
<th>Nexthop</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.2.3.4/24</td>
<td>192.168.200.2</td>
<td>IBGP</td>
</tr>
<tr>
<td>192.168.200.0/30</td>
<td>12.0.0.1</td>
<td>IGP</td>
</tr>
<tr>
<td>10.0.0.1/32</td>
<td>lspA</td>
<td>RSVP</td>
</tr>
<tr>
<td>12.0.0.0/30</td>
<td>-</td>
<td>direct</td>
</tr>
</tbody>
</table>

DMZ nexthop

<table>
<thead>
<tr>
<th>Route</th>
<th>Nexthop</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.2.3.4/24</td>
<td>10.0.0.1</td>
<td>IBGP</td>
</tr>
<tr>
<td>10.0.0.1/32</td>
<td>12.0.0.1</td>
<td>IGP</td>
</tr>
<tr>
<td>10.0.0.1/32</td>
<td>lspA</td>
<td>RSVP</td>
</tr>
<tr>
<td>12.0.0.0/30</td>
<td>-</td>
<td>direct</td>
</tr>
</tbody>
</table>
Next-hop self issue

• There are routes both from IGP and RSVP. Internal traffic should use the IGP, external should use RSVP. RSVP adds end-points in a separate inet table which BGP uses: inet.3. RSVP has lower precedence than the IGP. BGP looks in both inet.0 and inet.3. IGP does not. This is how transit traffic uses the LSP tunnels.