Label Switching

The idea

• Add a small label (sometimes called a “tag”) on the front of a packet and route the packet based on the label
How it works

When the packet reaches a router, the label acts as an index into a table of output lines, simplifying and speeding up routing,

Conceptually similar to a VC number (but setup is different)

Why?

• One original goal was faster switching – this is not as important now that IP routers are faster
• QoS for IP(and others) – routing over specific paths
• A label-switching router can route multiple protocols – in particular, IP can run on other network types
MPLS

- IETF calls this Multiprotocol Label Switching (MPLS) – RFC 3031

MPLS packet formatting (shim header)

- Shim headers are used for most non-ATM networks
- A “Layer 2.5” protocol?

Example: TCP over IP/MPLS

<table>
<thead>
<tr>
<th>Ethernet header</th>
<th>MPLS header</th>
<th>IP header</th>
<th>TCP header</th>
<th>App data</th>
<th>Ethernet trailer</th>
</tr>
</thead>
</table>

```
<table>
<thead>
<tr>
<th>Label</th>
<th>QoS</th>
<th>S</th>
<th>TTL</th>
</tr>
</thead>
</table>
```

20 bits 3 1 8
MPLS packet formatting (ATM cell header)

Position in cell normally taken by VPI and VCI

Label swapping

- Like VC’s, labels are not globally unique.
- Labels are swapped at each router

Incoming

<table>
<thead>
<tr>
<th>Output Link #</th>
<th>New Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>01 62</td>
</tr>
<tr>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>

Outgoing

<table>
<thead>
<tr>
<th>IP header</th>
<th>TCP header</th>
<th>App data</th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Route aggregation

- Flows that have the same destination are often given the same label by routers (the flows are “aggregated”)
- Flows aggregated in this way are said to belong to a single Forwarding Equivalence Class (FEC)

How Route Aggregation differs from conventional VCs

- If two flows were combined into a single VC, there would be no way of telling them apart at the destination
- Packets that are sent as part of an FEC still have their original TCP/IP headers, so they can be separated at the destination
Setting up MPLS forwarding tables

- Unlike VC’s there is no setup phase
- Forwarding tables are set up in one of two ways:
  - Data driven approach
  - Control driven approach

Data driven approach for setting up forwarding tables

- Common in ATM networks
- When a flow is first handled:
  - Router receiving a packet determines output link by standard routing
  - Router asks the downstream router to generate a locally-unique label
  - Performed at each router to the destination
Control-driven approach for setting up forwarding tables

- Common on non-ATM networks
- When a router is booted: I
  - It identifies which hosts it is directly connected to
  - Generates a local label for each host
  - Sends the labels to neighbor routers
  - Neighbor routers assign a new local label and send to their neighbors

Stacking labels

- Sometimes it is convenient to have “higher-level” virtual networks (e.g., VPNs or Overlay networks)
- Labels can be defined for the virtual network as well as the actual one
- A single packet can carry any number of labels.
- The “S” flag in the MPLS header indicates if this is the innermost header.
QoS support under MPLS

• With MPLS, we can direct IP packets to follow paths that would not be chosen by conventional routing (“explicit routing”):
  – High-bandwidth paths
  – Low-usage paths
  – Paths that use or avoid certain areas

Traffic Engineering

• MPLS’s support for explicit routing also allows network engineers to adjust the routing of flows to balance use of network resources
MPLS support for VPNs

• It’s simple to implement a tunnel with label switching – you just wrap the encapsulated protocol with MPLS headers
• This approach supports any common VPN technique

Pseudowire Emulation

• MPLS can be used to emulate another type of network (for VPNs, etc)
Pseudowire Emulation example: Emulating ATM over IP

- We want to emulate an ATM VC over an IP network:
  - Cells enter the VC on a particular input port with a particular VCI
  - They exit the VC on a particular output port with a (different) particular VCI

How to do it

MPLS labels implement the emulation by tunneling and also carry control info to the far end

ATM cell arrives

ATM cell sent

“Head” router

“Tail” router

Forwarded using TL

Demux label tells receiver that this is an emulated VCI

Tail router strips TL

Outer label implements an end-to-end tunnel

Inspects DL and passes to dest as ATM cell