9 Not just simple routing

9.1 Goal of this section

The previous two sections covered how internet scale IP routing works at the simplest level. This chapter looks beyond that to explain how that works in more detail. The goal is for readers to understand how we can use traffic engineering to make better use of networking resources beyond just using IP. Readers should understand how label switching works in contrast with IP routing. We'll also look in detail to understand how the simplest protocol for sharing label information used for label switching works: Label Distribution Protocol (LDP).

9.2 Problem: Use of resources

IP routing effectively uses SPF trees to enable routers to forward packets to destinations at different scales. However excellent your best route is, it is likely to get overloaded with traffic, while other "less good" links are under-utilised. Companies responsible for conveying traffic across the internet want to use all of their resources, not just some subset. How do we get more use out of our network? This is a (very lucrative) area of networking called traffic engineering.

9.3 Equal cost multi-path

The simplest thing we can do here that fits nicely within IP routing is called equal cost multi-path (ECMP). When two different routes are equally good, instead of tie-breaking in some way between them, we use both, and forward packets on both routes.

This is cheap to do. Some routing protocols (for example OSPF) do this by default, advertising and using two (or more) next hops for a destination if the route cost via each of them is the same. However, there are a couple of important limitations:

- As an administrator setting up path costs, you need to be very careful in construction of your link costs to ensure that the paths that you want to be equal cost are so (and the paths that you *don't* want to split traffic over are not).
- Packets are routed randomly, with 1/nth of the traffic over each of the n links. As this is split randomly, that's 1/nth of the traffic sent by every endpoint that is travelling over that link, so if one of your links goes down, every endpoint is affected.
- Similarly, ECMP provides an equal split between the available routes – so if your network links have different bandwidths, it's not possible to (say) split your traffic 80/20 between them to reflect the different bandwidth available

9.4 Label Switching

Label switching is a technology that allows us to create virtual IP links in our IP network, where the actual route that the data takes is hidden from the IP routing. In label switching parlance, these are called **tunnels.** As far as your IP routing is concerned, your traffic vanishes into one end of the tunnel, travels invisibly, and pops out the other end somewhere else.

This section will explain how label switched tunnels work, and how we can use this to create a label switched network that makes better use of our resources.

A note before we start. Label switching breaks the OSI layering model, running at layer 2.5. The data forwarding down tunnels is at layer 2 (and is indeed called switching), but it works at layer 3 scale and the protocols that control what data is switched how/where run at layer 3.

9.4.1 Label Switching Abstract

Conceptually, label switching works as follows:

- A stream of traffic entering a network is tagged with a label. Streams of traffic can be identified or defined in different ways, but crucially not just by destination. For example, all traffic from endpoint A to big-server B could be tagged with label X, but all traffic from endpoint Y to big-server B could be tagged with label Y (and thus treated differently). A stream of traffic that is all to be forwarded in the same way is called a **forwarding equivalence class (FEC)**
- Every router in the network is programmed to match incoming data based on the combination Incoming interface and label, and map that traffic to an outgoing interface and label (which may or may not be the same label). Thus each stream can be treated independently, but all traffic in a particular stream is switched consistently across the network.
    - Note: instead of routers and routes, these are **label switch routers (LSRs)** and **label switched paths (LSPs).** Routers at the edge of the label switched network that perform the job of identifying streams of traffic and adding labels – and removing labels from traffic leaving the network – are called **label edge routers (LERs)**
- As the traffic leaves the label switched network, the label is removed, and the stream of traffic can resume IP routing (or whatever it was doing before it was interrupted).

The above provides a point-to-point tunnel (actually potentially a whole set of point to point tunnels) that depending on the label rules programmed on the LSRs can take any route through the network.

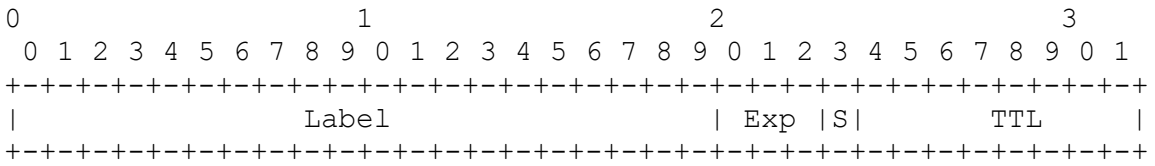
This gives us two really useful abilities.

- By labelling different streams of traffic differently, we can send the traffic different ways – making fuller use of the network. Because we can even label different streams of traffic between the same source and same destination differently, we can really break down the load on our network and make very good use of it.
- By creating and programming in two node-disjoint LSPs for the same stream of traffic between two LERs, we can have a primary and backup route for the same tunnel, allowing redundancy to failure within the scope of a tunnel. If the LERs spot that a particular stream of traffic has stopped, they can swap to the backup LSP that is using a different part of the network.

9.5 Multi-Protocol Label Switching (MPLS)

9.5.1 MPLS Protocol Header

The standard implementation of label switching used in the internet is MPLS. The MPLS label header fits between the Ethernet (Layer 2) and IP (Layer 3) headers. It is also called the Shim header.



This consists of the following fields

- **Label** (20 bits). The label attached to a stream of traffic. This is the label used in combination with the incoming interface for the switching rules. It is unique on a link within the label switched network. Remember that streams are switched based on interface *and*

label, so it is not necessarily unique across the whole network, just for the link. That said, it is nice not to have to rewrite labels, so these are often unique across a network.

- **Experimental bits** (3 bits). Some bits reserved for future enhancements. We'll ignore these in this course.
- **Bottom of stack bit** (1 bit). This is set to 1 on the last MPLS header in the stack, indicating that something other than another MPLS label (usually the IP header!) comes next.
- **Time to Live** (8 bits). Works in the same way as an IP header. The TTL of the top label is decremented every time it is switched.

9.5.2 Layering

MPLS allows layering, allowing LSPs to be nested. This is useful, for example in large networks where there are a large number of LSPs which all traverse the middle of the network in the same way. Rather than having a huge (and slow) forwarding table lookup on every LSR in the middle of your network, you can put them all down the same LSP (with one new label) which can itself be switched.

9.5.3 Example Flows

09_MPLS_Simple.pcap and **09_MPLS_Multiple_Labels.pcap** show MPLS headers on packets travelling down an LSP. These show someone using the ping tool, sending pings down the LSP and with the replies coming back via raw IP.

9.6 Label Distribution

With IP routing, we rapidly discovered that it would be painfully slow and complicated for an administrator to manually configure routes on every router in the network. Similarly, it would be just as hard, or worse, for an administrator to manually configure every label to be used on every link. Just as IP routing protocols solve this problem for IP, so we have label distribution protocols that handle distributing labels.

9.6.1 Label Distribution Protocol

The simplest label distribution protocol is (unimaginatively) called Label Distribution Protocol (LDP). LDP runs over IP – the LDP protocol messages themselves travel through an IP network called the **control channel**. LDP enables LSRs (Label Switched Routers) to set up label switched data paths for data forwarding (the **data channel**) that are not IP. The control and data channels might both run over the same Ethernet network, or might not. Note that although LDP is the first time we've come across this, the concept of having a separate control and data channel is not specific to LDP.

LDP does three things:

- Discovers other neighbouring LDP-capable LSRs that are neighbours (similar to OSPF)
- Establishes and maintains control adjacencies (called LDP sessions) with those neighbours (similar to OSPF)
- Requests, advertises and withdraws labels to create one-hop LSPs for FECs (Forwarding Equivalence Classes).

9.6.2 Neighbour discovery and session establishment in LDP

LDP broadcasts "Hello" messages out each interface to the "all routers on this subnet" well known multicast address 224.0.0.2 to port 646. Those Hellos include the address that the LDP router would like to use to establish an LDP session with its neighbours (not necessarily the address that the Hello messages are sent from). LDP listens on 224.0.0.2, port 646, and discovers the presence of neighbours (and their transport addresses) due to incoming Hellos – and then establishes a point-to-point LDP session directly to each neighbour.

LDP routers can also be explicitly configured with neighbours and will (try to) establish sessions with those configured neighbours without waiting for Hellos.

The unicast session establishment includes negotiation of parameters such as timer values and what range of labels they will use. Both endpoints run a keep alive timer, and if they do not receive any message from their neighbour within that time, they assume that the session has gone down.

9.6.3 Label distribution in LDP

Label distribution in LDP can be proactive (called **unsolicited**) or on request (**downstream on demand**).

In unsolicited mode, LDP simply informs its neighbours of labels to use for any FEC that it is aware of. In downstream on demand mode, LDP waits to be asked for a label for a particular FEC.

What is a "FEC" in this case? Firstly, any destination in the IP routing table on an LSR that is acting as an LER. Secondly, any LSR that receives a label on an interface that could use that destination as a FEC – advertising a label upstream that matches a particular label downstream (stitching 1-hop LSPs together).

9.6.4 Example Flow

09_LDP.pcap shows an LDP session being set up. In this capture you see the following:

- Two LDP routers at 10.0.0.1 and 10.0.0.2 are sending out LDP hellos, advertising transport addresses 10.0.1.1 and 10.0.0.6 respectively
- The LDP routers start a TCP connection between 10.0.1.1 and 10.0.0.6 and start an LDP session over it. See packet 17 for the first LDP initialization message sent to establish the session. Again, don't worry about the details of the TCP messages – we'll cover later in the course and come back to this.
- The routers exchange label mapping information on that session. See packet 21 for an example.
- Although the routers have no further information to exchange, note that they continue to send keep alive messages (for example packet 47) every so often to prevent the session timing out.

Note: This flow is to help understand how LDP works at a high level. You do not need to know the details of the LDP protocol messages and flows for this course.

9.6.5 Constraint-based LDP

Constraint Based LDP (CR-LDP) also allows the specifying of explicit routes - lists of LSRs (or groups of LSRs) that a particular LSP must travel through, and allows LSPs to be bandwidth limited.

The specifying of explicit routes requires Downstream on Demand labels, but allows source LSRs to specify a list of hops through a network for a particular LSP.

9.7 Traffic Engineering: Guaranteed Bandwidth and Backup Provision

Once we have LSPs with known bandwidths and traffic engineering (source routing LSPs) in a network we have a lot of control over what goes down each link in our network. If we can then get the live information on currently reserved/in use bandwidth spread around the network, then the source LSRs know what the available "bookable" bandwidth in the network is. If they know that, then they can create new LSPs that don't overload the bandwidths available on the existing links. This means the network can guarantee bandwidth allocation through their networks for particular source-destination paths.

Being able to provide guaranteed bandwidth to customers is very valuable (consider companies or more critically emergency services internet usage!)

How do we get the “currently used bandwidth” information around? Link state routing protocols already flood topology information around the network for routing calculations and can easily be modified to provide this extra info. Traffic engineering extensions exist for most link state routing protocols to enable this (for example OSPF-TE). We won’t cover the details of how these work in this course.

Similarly, LSRs can calculate not just one but two disjoint LSPs through the network between a source and destination (either link disjoint, or node disjoint) to provide a pre-configured backup (also guaranteed bandwidth), providing an even safer link across the network.