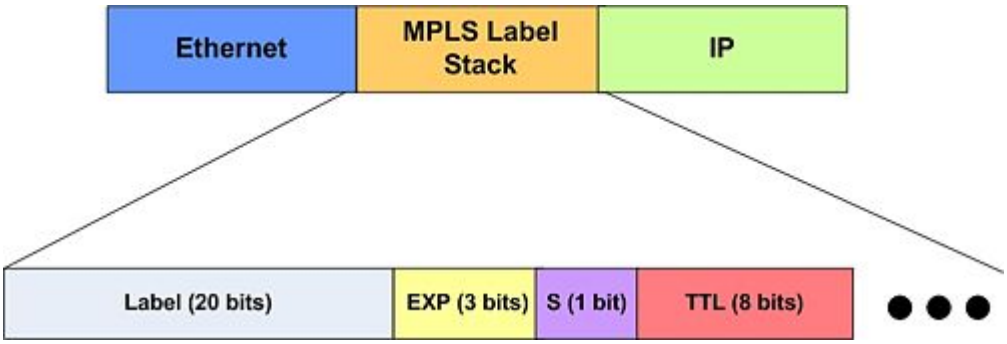
MPLS training

MPLS HISTORY

- Developed by Cisco as TAG SWITCHING. Many commands on some of the mid-2000s routers still referenced tags rather than labels, despite the ratification of the standard much earlier than this (IETF turned it into an open standard)
- Original goal was for high speed switches/routers that would only need to see a label and know what to either change it to, or remove it and forward. No longer much of a factor on most kit, as a lot of forwarding done in hardware rather than software

MPLS LABEL



Label Stack contains the actual label itself (max of 20 bits), the EXP field, the S field, and a TTL. TTL works like TTL in IP. The S field denotes whether this is the last label in the stack (it could be the top label, with VPN labels or otherwise under it). EXP bit is the experimental bit, usually used for QoS matching. 8 distinct values, just like ToS/CoS in IP.

Sits between Ethernet header and IP header, so IP lookup not required to forward packets.

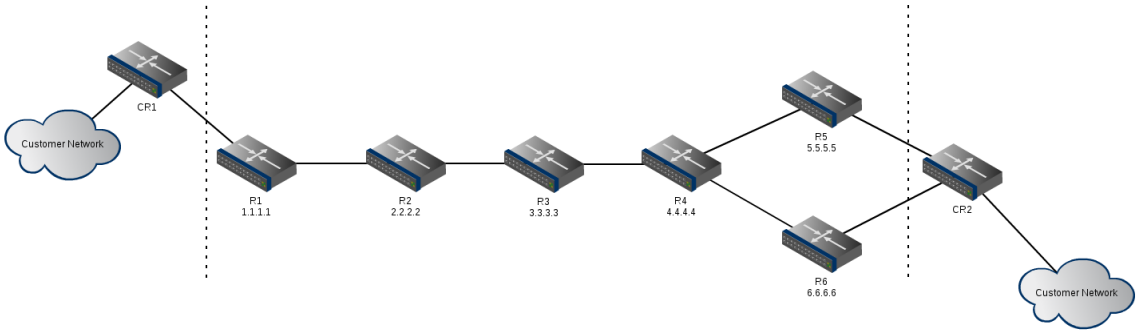
MPLS LABEL OPERATIONS

Three general operations for label switching, push, swap and pop

- Push - Add a label onto a standard IP packet
- Pop - Remove the label from an MPLS packet to put it back to how it was (usually IP under, unless more labels involved)
- Swap - Remove the label on the stack, but replace it with a new one. Most common operation, as this is generally how packets are forwarded through an MPLS network

Due to the fact that nothing above the ethernet layer is seen by the transit network, practically any Layer 3 (and even Layer 2) protocols can be encapsulated by MPLS. This includes IPv4, IPv6, Layer 2 VPNs, and many others

MPLS NETWORK



In the above network, there are multiple routers. CR1 and CR2 are customer routers. R1 to R6 are the service provider routers.

MPLS defines three different terms for the routers in an MPLS network:-

- Customer Edge (CE) - This is the router which will participate in routing with the Service Provider, which sits on the customer site. The customer will connect all connections they want into the SP network via these routers
- Provider Edge (PE) - This is the router which does routing exchanges with the customer, and also usually is the place where an IP packet gets turned into an MPLS packet (by **push**ing a label onto it)
- Provider (P) - These are the intermediate routers which do nothing but forward on MPLS frames.

MPLS LABEL DISTRIBUTION

For routers to forward on MPLS packets correctly, they need to know firstly what routers are in the network, what routes exist, and finally what each router has as a label for a particular route. The most common protocol to achieve this is called LDP (Label Distribution Protocol).

Firstly, all routers in the SP network must be running an IGP (OSPF, Integrated IS-IS, or whatever else). This means that all routers know about each others routes. LDP itself cannot run without an IGP present.

Secondly, a loopback needs defining, as LDP will not start without one. This needs to be reachable by all routers that will take part in the MPLS network, and will be used as a basis for next-hops. This becomes known as the transport address.

Once these steps are done, you can enable LDP on all interfaces you want to run MPLS on. You wouldn't tend to do this on any customer facing interfaces, but anything facing into the SP network you would.

What LDP does is distributes labels for next hops of routes. There is no need to have separate labels for multiple routes that go to same final router in the MPLS network, they can be forwarded exactly the same. LDP populates a "forwarding information base" so that packets can easily reference this and be forwarded accordingly.

The way this tends to work is that each router will look at it's routing table, work out all the potential next hops, and then choose a label for each next hop. It will then advertise these downstream. Any which it has received from the upstream will form part of its swap operation.

An example of this would be R4 having a route via R6, and thus it needs a label for the next-hop of 6.6.6.6. It may assign a label of say, 10 for 6.6.6.6. R3 will have already assigned a label to 6.6.6.6, for example, 20. When an MPLS packet that is destined for R6 comes in to R3 with the label of 20 (as this has been advertised downstream), it will remove the label of 20, and put a label of 10 on, to forward to R4. This is effectively all P routers should be doing (which in this case are R2, R3 and R4), taking MPLS packets in, swapping labels, and forwarding.

One thing to note is that LDP works on directly connected interfaces, so neighbourships are established, labels passed etc. If an LDP session is needed to a non-directly connected router, Targetted LDP will be used. It works in the same way as LDP does, except you are setting up what is effectively a tunnel between the source and

destination of the Targetted session to then distribute labels via.

Once all labels are distributed, LSPs are created. This is a Label Switched Path, and is nothing more than a path for traffic to take through the label switched network. Traffic that matches this Label Switched Path (i.e. has a next-hop which goes down this LSP and usually a similar source) is called an FEC, or a Forwarding Equivalence Class. This could be all traffic from one router destined to go to the same next-hop. Traffic from a different customer router may come in and use the same Label Switched Path (i.e. go to the same next hop), but it isn't going to return to the same router, so is part of a different FEC.

Penultimate Hop Popping

There is a behaviour called PHP (Penultimate Hop Popping) which lightens the burden for the Provider Edge router. When a packet reaches the final hop before it gets to the actual Provider Edge router, it would forward on the packet to the PE, which would then have to do a label lookup (which will tell it to remove the label anyway) and then do an IP lookup to then forward it on.

Given that the final P router will know to forward this on to the PE router, it makes sense to remove the MPLS label at this point (as it already knows where its going at this point), so that the PE router just has to do an IP lookup. This may not seem much on a per packet basis, but getting to multiple gigabyte speeds, this can alleviate CPU cycles/hardware forwarding processing time.

WHY USE MPLS?

In the early days, there was a performance benefit to using MPLS. A simple label lookup, rather than a full IP lookup was used, and this did save time, and meant that intermediate routers can have very little actual configuration on them and do little other than forwarding. However, most switches and routers do IP forwarding in hardware using cached routing entries, Cisco's CEF (Cisco Express Forwarding) being a good example of this. MikroTiks are an example of a router that don't forward in hardware, so MPLS does actually give a performance benefit on these.

Most of the reason to use MPLS now is for the applications/featureset that can be taken advantage of. The big three are Layer 3 VPNs, Layer 2 services (VPLS or VPN), and MPLS Traffic Engineering. In all cases, a new label is added specifically for this service, so that the recipient router knows what to do with it.

MPLS Layer 3 VPNs

Layer 3 VPNs are effectively giving a customer a private routing space/table which can span large geographical areas. VRFs themselves (Virtual Routing and Forwarding) are a way of keeping separate routing tables within a device, however to get these routes to different locations in the network, L3 VPNs are the most commonly used way.

They work on the basis that a customer advertises its routes into the Provider Edge router. The Provider Edge router stores these in a VRF specific to that customer. These routes are then advertised out into the MPLS network, and any router which also needs to have these routes in that VRF will import these routes. It's useful to know that every router in the MPLS network will actually carry the Layer 3 VPN routes, it is more a case of whether they import them and therefore use them, or just ignore them.

Route Distinguisher

The good thing with these is that the customer can run any range whatsoever, and so long as there is a different distinguisher on the PE router, they will not overlap with other customers. This is where something called a Route Distinguisher comes in. This is an 8 octet field, which tends to be in the format of Type (so L2VPN, L3VPN), ASN and a unique value. The type is not configured (the router will take care of this), so the format of these is usually configured like 41230:12 or 41230:13.

The resulting route for the customer would then be advertised out as the normal 4 octet IPv4 address (or 128-bit address if IPv6), along with the 8-octet Route Distinguisher. This means that two customers could use say,

192.168.1.0/24 and not overlap throughout the network.

Route Target

Not all routers are going to have every customer connected to them. You don't particularly want a router in Manchester to have the routes for a customer when they only have connections in London and Sheffield. While it wouldn't cause any problems, it isn't really necessary. This is where Route Targets come in. The idea is that routes in a VRF will be exported/advertised out onto the MPLS network with a Route Target, and then any router that requires these routes to be part of a VRF it has (say, a customers second location) can import any routes that match this Route Target. They are defined in a similar fashion as Route Distinguishers, with the format of ASN:ID, so you could use the same Route Distinguisher as Route Target (41230:12, 41230:100 etc)

Where does MPLS come into this?

None of the above is really specific to MPLS itself. Given MPLS is just for label switching, this doesn't mean a great deal as it is. Where MPLS comes in is that each of the VPN routes are given a label. This is on top of the existing label use for forwarding (to the next-hop, as defined by the IGP/LDP). When the MPLS label for forwarding through the network is removed, this leaves the VPN label. The router then looks in it's forwarding table to see if it can match against the label in the received packet, and then forward it out to it's destination based upon that.

As mentioned, every single VPN route has a label, which needs to be advertised out. This is done using MP-BGP (multiprotocol BGP) which not only advertises the routes, but also the labels associated with that route. This means that a sending router will always know that to reach a certain route in the VRF, it will add a label onto it. The labels also provide a quick lookup too, in the same way labels do for forwarding packets.

As an addition, all routes in a VRF have a specific Route Target, which is advertised as an "Extended Community" in the BGP route. Communities can be used for all sorts of purposes. They are an arbitrary value, which the recipient of the community can decide what to do with. This is based upon matching an extended community (in the same way you'd match an IP address in an access-list). You can then set an action based upon that. A common use of this is to send routes out to a service provider with a community value on, and they set something like Local Preference or other metric values, so that routing takes the path preferred by the customer, rather than left to either the SP to decide, or the IGP/routers themselves.

While other methods could be used to achieve this, MPLS with BGP and its ability to carry extended attributes is a very efficient and easy way to do this.

VPLS and Layer 2 VPNs

A VPLS is the act of extending a layer 2 connection across multiple geographical locations. There are two ways to go about it. One is statically defined, and setup via LDP, with a unique VPLS ID defined for each Layer 2 domain. This requires setting up VPLSs to every router that needs to be a part of it (so if between 3 routers, explicit configuration is required on each router to have one VPLS which has Layer 2 endpoints on each).

Another way of doing them is via BGP-signalled VPLS. This is done in a similar way to Layer 3 VPNs, in that a route-distinguisgher is defined, and route targets are too. As the Layer 2 VPN routes are on every single MPLS router in the network, all it takes is importing and exporting the route targets on the local router, and BGP will signal to the other routers with the same import and export route targets to set up a Layer 2 connection to this router. BGP then adds labels for each VPLS endpoint and advertises these out to all routers which take part in this VPLS

What you would do is then bind either an interface, or a group of interfaces to this VPLS. This then looks like a one big flat Layer 2 switch to the customer at this point, with the ability to forward anything used at layer 2. This includes VLAN tagging, standard Layer 3 endpoints (making the VPLS look like a big leased line), and even protocols like Spanning Tree.

The Layer 2 frame itself gets encapsulated by an MPLS label (as signalled by either LDP or BGP), as well as

another Layer 2 header. This Layer 2 header is not the same as the one that has been encapsulated, it is actually just a standard Layer 2 header for forwarding the frame through the Service Provider network, in the same way any standard frame (whether containing IP, MPLS, or otherwise) gets a Layer 2 header.

MPLS Traffic Engineering

MPLS Traffic Engineering is a useful technique in that rather than forwarding traffic at a best-effort way, or following the IGP path blindly, it can actually take advantage of other attributes (bandwidth, administratively-preferred links etc). There are multiple protocols which could be used for this, but the most commonly used (and more often than not the only ones implemented on many routers) are RSVP, OSPF, and a similar protocol called CSPF (Constraint-based Shortest Path First). CSPF and OSPF work in conjunction with each other.

OSPF usually does little other than create LSAs to flood out to all routers in the OSPF domain, to detail networks and where they are located. When using MPLS TE however, OSPF has extensions which allow it to also pass along the guaranteed bandwidth of each link (defined by an administrator usually), whether an administrator has defined a link as preferred even if it is not the shortest/IGP-chosen best path, and other details. These are then fed into a protocol called CSPF, which goes through the same Shortest Path Algorithm (Dijkstra algorithm) as OSPF does, but takes into account bandwidth constraints, the administrator chosen links and the like, which then forms what is the best path for traffic to take.

However, there is something missing in here. While OSPF and CSPF can put together the constraints defined on each link, they have no way of changing these constraints. This is where a protocol called RSVP (Resource Reservation Protocol) comes in. This protocol will first set up an MPLS Label Switched Path, but also will have a bandwidth reservation required for this path. If the bandwidth is available on the interfaces is traverses, it takes this path, and then decrements the available "resources" on the interfaces it passes through. If not enough resources are available, it will notify OSPF and CSPF on the router which attempted to establish this path (also known as the head-end router), which will recalculate CSPF based upon what resources are required.

Once RSVP has set up this MPLS LSP, it will add MPLS labels for traffic to forward based upon, which gets the traffic through the network.

MPLS TE is usually defined as a tunelling protocol, because effectively the LSP is set up by it, and it encapsulates traffic with a label based upon resources available. Due to it setting up its own path and encapsulating the packet, it is a tunnel in all but strictest definition. In many implementations, tunnels are actually created in the configuration to use for traffic that is to be subject to traffic engineering. These then have mechanisms to attract down this tunnel (whether by static routing that traffic down this tunnel, or by telling OSPF to see this tunnel as a preferred interface over the current IGP route for certain traffic). Another thing to note is that priorities can be set, so that certain traffic can be given priority over others. This could be useful in an outage scenario, where core traffic needs a guaranteed resource availability, so giving this a higher priority over customer traffic (or even something like TV/streaming services) are booted off a certain path, the core traffic takes the resources, and the customer/streaming traffic has to find a new path.

It is possible to also tell an MPLS TE Tunnel to have a manual path. Usually with RSVP, OSPF and CSPF, this is taken care of automatically. However, there could be a situation where a strict path may be preferred and can be explicitly set (by defining each hop to take), or it could be a backup method to the automatic tunnelling. A good example of this being useful is similar to the previously mentioned one, in that forcing customer traffic to run down a defined path when all automatic paths have their resources taken up by higher priority traffic, which could then be rate limited, or simply via a less preferred route around the network.

Conclusion

MPLS is a very simple protocol to set up, it can actually be enabled in a couple of commands on many routers. The applications that run on top of it provide the actual complexity and where the thought needs to be when creating an MPLS network. Understanding the technology is way more vital than learning the commands, as the commands are very simple, but very powerful. Placing MPLS TE in the wrong place or using the wrong constraints (bandwidth, priority etc) could mean web traffic takes a suboptimal path round the network, whereas

someone's streaming traffic gets the best path. By the same token, MPLS VPNs turning up in the wrong place, or not being imported/export at all and thus disconnecting customers edge sites from each other can obviously lead to problems.

For any further questions, please talk to me