Internet Technology

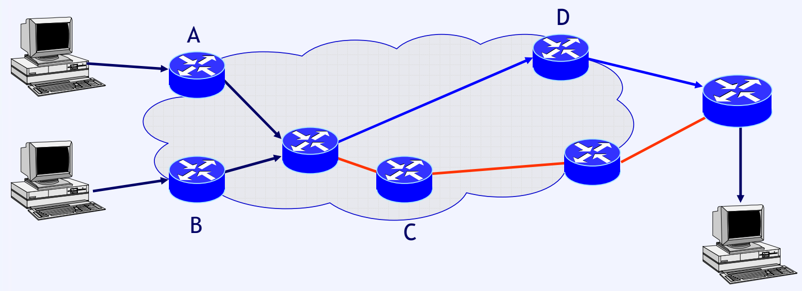
The "inner network" view, part 2: MPLS

Michael Welzl   http://www.welzl.at

DPS NSG Team http://dps.uibk.ac.at/nsg
Institute of Computer Science
University of Innsbruck, Austria

Traffic Engineering

- Static configuration: administrators want to move some traffic
- based on long-term measurements



- IP-in-IP tunneling example:
- B encapsulates packets (new src=B, dst=C), C removes new header

From traffic engineering to MPLS

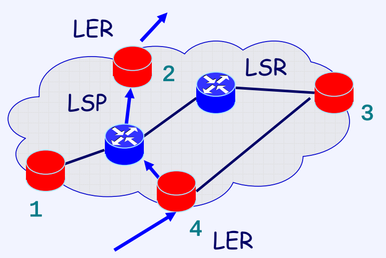
- Layer violation:
  - If you are tunneling along a fixed path and your network is ATM, you could just as well set up a VC for the path - faster forwarding!
- Automatic variant: Ipsilon IP Switching
  - Switches identify flow (MF classification), establish ATM-VC "Short-Cut"
  - Does not scale well - fine granularity
- Better: Multiprotocol Label Switching (MPLS)
  - Not just (but mostly) ATM anymore...
  - based upon separation of forwarding and control functionality in routers
  - Label Edge Routers (LERs) put short info. (from layer 2) in front of IP
    - like IP encapsulation, just not with a whole outer IP header
  - Label Switching Routers (LSR) forward on Label Switched Path (LSP)
  - At destination: remove label, forward IP packet normally

MPLS tunnels

- Efficient tunneling is the key functionality of MPLS
  - Tool for efficiently connecting edges
- Essentially, **MPLS adds connection orientation to IP!**
  (and as such, has a clear control plane / data plane separation)
  - Yes, connection oriented IP goes against some fundamental principles
  - So many people hated it, and there were long and heated debates
  - In the end, the market gave MPLS the thumbs up
- Some features of MPLS tunnes:
  - Traffic can be explicitly routed
  - Recursion: build tunnels inside tunnels inside tunnels
  - Protection against data spoofing (only the head of a tunnel can inject data into a tunnel)
  - Low encapsulation overhead

Roles of tunnel endpoints vs. LSRs

- Like in DiffServ: job made easier for core routers
  - Why even do "normal" routing? (routing updates, memory + maintenance effort for tables)



- Example on the right
  - Assume that traffic is always routed either vertically or horizontally; then left inner LSP's "routing table" becomes: From 1 to 3; From 3 to 1; From 2 to 4; From 4 to 2
  - This table is called Next Hop Label Forwarding (NHLF) table
    - Maps FEC (if not LER: given by incoming label) to a set of operations
  - Less state = more scalable
  - LSRs do not need to consider IP header ⇒ more efficient

Some MPLS Applications

- Traffic Engineering (TE)
  - Influencing where traffic goes for the sake of better resource usage
- Layer 3 VPN
  - Also known as IP VPN
- Point-to-point layer 2 transport
  - Carry a customer's Ethernet traffic across a WAN
  - Component of ATM or Frame Relay service emulation
- Virtual Private LAN Service (VPLS)
  - Customers of an ISP are given the impression of sharing a LAN
- Network convergence: save money by connecting services from distinct networks instead of building a new network
  - e.g. Public Switched Telephone Network (PSTN) + Internet + ATM + Digital TV...

⇒ MPLS = Key enabling technology for many things, not just TE!

Label format

- Forwarding Equivalence Class (FEC)
  - Group of packets with similar expected treatment (usually same label because usually same destination)
  - Various forms of classification possible (MF, ..)
- What if labeled packets are labeled again?
  - Labels are stacked (push, pop, swap [= pop+push])

20 bit	Label	Label	Label	IP Header	Data
3 bit	Experi-mental	Experi-mental	Experi-mental		
1 bit	S=0	S=0	S=1		
8 bit	TTL	TTL	TTL		
	Stack top		Stack bottom		

MPLS encoding

- Label is not what link layers expect when carrying IP
  - MPLS supposed to be "multi-protocol" below as well as above
- Specifications for carrying label needed for link layers
  - ATM: label contained in VCI/VPI field of ATM header
  - Frame Relay: label contained in DLCI field in FR header
  - PPP/LAN: uses 'shim' header inserted between L2 and L3 headers
- How is a label detected by the link layer?
  - RFC 3032: *"The ethertype value 8847 hex is used to indicate that a frame is carrying an MPLS unicast packet."*
- Such a field does not exist in the label - so how to detect the network layer protocol (e.g. IPv4 vs. IPv6)?
  - Configuration: associate label values with network layer protocol or use it only for one protocol (e.g. only IPv4 everywhere)

MPLS details

- Label designed for speed:
  - 32 bit
  - S=1: "this is the last label"
  - TTL is the only IP header field that must be treated at each hop
- Normal operation: one label per link
  - Ingress LER
    - identifies egress LER + corresponding LSP
    - applies label value corresponding to LSP (push)
  - Next routers along LSP
    - performs lookup of label
    - determines and applies output label (swap)
  - Egress LER
    - removes label, forwards as a normal IP packet

MPLS operation

- Intermediate routers need to swap labels
  - Not always necessary; in simple configurations, same label can be kept
- Egress LER carries out two tasks
  1. remove label
  2. IP routing

  ⇒ Common simplification: Penultimate Hop Popping (PHP) (penultimate router pops label)
- Why stack labels?
  - Create LSP tunnel within LSP
  - e.g. to differentiate between two VPNs:
    - use inner label to identify service
    - use outer label to quickly send packets through ignorant routers (where differentiation is unnecessary)

MPLS and DiffServ

- PHB must be determined via label
  - EXP(erimental) bits
- Two methods
  - E-LSP (EXP-inferred LSP): map EXP ⇔ DSCP
    - Up to 8 different PHBs possible
    - Packets requiring different PHBs transmitted on same LSP (but different queues)
    - Not signaled when establishing LSP, but statically configured
  - L-LSP (Label-inferred LSP): map EXP+label ⇔ DSCP
    - PHB number not limited by MPLS
    - Possible to use different LSPs for different PHBs
    - Must be signaled when establishing LSP (as labels are tied to LSP)

The MPLS Control Plane: LDP

- How to configure MPLS?
  - Special protocol needed
- Community could not agree whether to extend existing protocols or design a new one...
  - So they did both
  - Result: Label Distribution Protocol (LDP) + RSVP, OSPF, BGP extensions
- LDP uses TLVs (Type-Length-Value triplets)
  - Encoding begins with TL, length of this field known
  - V content and size can vary
  - TLVs facilitate
    - adding new capabilities (define new type)
    - skipping unknown objects (just look at TL, ignore V)
- Side note: penultimate hop popping requested by egress LER by advertising "implicit-null" label (special defined value 3), which means "just pop, please"

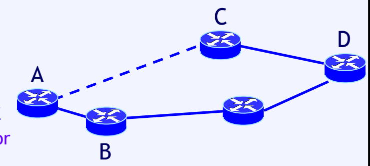
What is underneath LDP?

- LDP finds peers via multicasting UDP "hello" messages
- Then initiates TCP connections to peers, set up LDP session
  - Initially: exchange information about features and operation modes
    - Downstream Unsolicited (DU) vs Downstream on Demand (DoD)
    - Both methods can coexist in the same network
  - Then, exchange Label ⇔ FEC mapping
    - Messages for mapping, withdrawing labels etc.
  - Maintain connection for incremental updates
  - TCP does not guarantee that silent peer is still there ⇒ keepalives added
- LSPs which are set up by LDP use IGP
  (*"LSR A that receives a mapping for label L for FEC F from its LDP peer LSR B will use label L for forwarding IFF B is on the IGP shortest path for destination F from A's point of view"*)

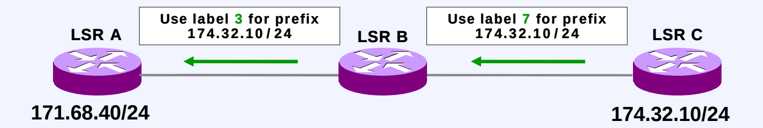
LDP - IGP

- The good: relying on IGP means that
  - IGP helps LDP prevent loops
  - IGP path change affects LSP (i.e. LSPs are self-healing :-))
- The bad: it also means that
  - LDP LSPs cannot traverse autonomous system (AS) boundaries (IGP scope)
  - Reconvergence time of IGP = lower bound on LDP reconvergence time
    - During reconvergence, traffic may be blackholed or looped (normal for IGP - LDP inherits this property)
    - LDP-IGP-synchronization problems can lead to race conditions
- Example synchronization problem to the right:
  - Assume path A-C was unavailable during LSP setup
  - Assume it becomes available later, and IGP reacts faster than LDP
  - Due to rule on previous slide, A stops using the binding it received from B, and LSP stays down until A receives a binding for the same FEC via C
  - Possible solution: advertise high IGP link costs for links when they do not have an LDP session

Note: such loss of synchronization can also be caused by firewalls, misconfiguration, ..



Downstream Unsolicited (DU) Distribution



LSR A:

In I/F	In Lab	Address Prefix	Out I/F	Out Lab
0	-	174.32.10	1	5
...	...	...	...	...

LSR B:

In I/F	In Lab	Address Prefix	Out I/F	Out Lab
0	5	174.32.10	1	7
...	...	...	...	...

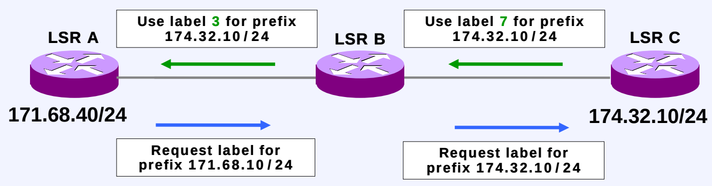
LSR C:

In I/F	In Lab	Address Prefix	Out I/F	Out Lab
0	7	174.32.10	1	-
...	...	...	...	...

IGP derived routes

- LSRs assign a label to each FEC
- LSRs distribute labels to the upstream neighbours
- Disadvantage: unnecessary signaling traffic

Downstream on Demand (DoD) Distribution



- LSRs assign a label to each FEC
- Upstream LSRs request labels to downstream neighbours
- Downstream LSRs distribute labels upon request
- Disadvantage: after LDP-IGP synchronization problem, LSR can only be repaired when a new request was satisfied
  - Significant delay

Label assignment and retention mode

- Downstream label assignment: router expects to *receive* traffic with the label that it picked locally
  - Traffic flows in opposite direction from distribution of labels
  - E.g., LSR A receives label L1 for FEC F and advertises label L2
    ⇒ traffic for FEC F should arrive with label L2, label L1 will be applied
  - "downstream" because next-hop label was picked by downstream router (in traffic direction)
- Label retention mode
  - Consider: LSR A which receives unsolicited label advertisements:
    - for FEC F with label L1 from peer B
    - For FEC F with label L2 from peer C

    ⇒ what should LSR A do?
  - Depends on policy: liberal or conservative label retention mode
  - E.g. liberal:
    - LSR A puts label L1 in forwarding table but remembers L2
    - If IGP path changes and points to peer C, LSR A simply replaces L1 with L2

Liberal vs. Conservative Label Retention

Liberal	Conservative
• LSR maintains bindings received from LSRs other than the valid next hop	• LSR only maintains bindings received from valid next hop
• If the next hop changes, it may begin using these bindings immediately	• If the next hop changes, binding must be requested from new next hop
• May allow more rapid adaptation to routing changes	• Restricts adaptation to changes in routing
• Requires an LSR to maintain many labels	• Few labels must be maintained

Label retention method trades off between label capacity and speed of adaptation to routing changes

Ordered vs. Independent LSP Control

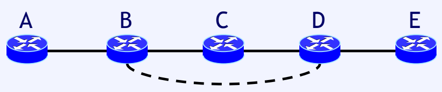
Ordered	Independent
• LSR only binds a label to a particular FEC if it is the egress LSR for that FEC, or if it has already received a label binding for that FEC from its next hop for that FEC • Ordered LSP setup may be initiated either by the ingress or the egress	• Each LSR, upon noting that it recognizes a particular FEC, makes an independent decision to bind a label to that FEC and to distribute that binding to its label distribution peers • Communicate FEC - label binding to peers once next-hop has been recognized • LSP is formed as incoming and outgoing labels are spliced together

- Both methods supported in the standard and fully interoperable
- Both have their pro's and con's ...

Ordered vs. Independent LSP Control /2

- Ordered LSP control
  - Needs more delay before packets can be forwarded along the LSP
  - Depends on availability of egress node
  - Consistent granularity, avoids loops
  - Used for explicit routing and multicast
- Independent LSP control
  - Labels can be exchanged with less delay
  - Does not depend on availability of egress node
  - Granularity may not be consistent across the nodes at the start
  - May require separate loop detection/mitigation method
- E.g. consider routing change:
  - Ordered control: labels must propagate to routers in the new IGP path
    - But can be sent along with IGP messages themselves
  - Independent control: labels are already there

Ordered vs. Independent LSP Control /3



- Example topology above: LSP established from E to A (traffic flows from A to E)
  - direct B-D line added later by operator who includes it in IGP but forgets to enable LDP on it
- Ordered control
  - B notices that label advertisement for FEC E from LSR C ≠ IGP best path
  - B withdraws its advertisement for FEC E, removes forwarding state
  - A receives withdrawal, removes forwarding state, knows that LSP for FEC E is not operational, will not attempt to use it
- Independent control
  - B notices that routing changed, and outgoing label in forwarding table for FEC E is no longer valid ⇒ removes forwarding state for FEC E
  - A does not change forwarding state (B is still on best path for A)
  - Result: LSP is broken at B, but A is unaware of failure; problematic e.g. for VPNs
  - Again, possible solution: advertise high (IGP) link costs when link does not have an LDP session

LDP Summary

- Key features
  - Automatic discovery of peers
    - ease of configuration
      (no need to manually update existing LSRs as new ones are added)
    - Amount of session state at an LSR proportional to number of neigbors
  - Reliable transport
    - Because TCP (+ keepalives) used for all messages except discovery
  - Extensibility
    - TLVs
  - Reliance on IGP
    - Has its good and bad sides...
  - Liberal label retention and downstream unsolicited label distribution
    - Labels are advertised to all peers and kept by peers even if they are not actively used for forwarding ⇒ LDP can quickly react to routing changes
    - Alternative: Equal Cost Multi-Path (ECMP)
      multiple forwarding table entries for load balancing