NSWI184 – Řízení počítačových sítí

Přednáška desátá

Ondřej Zajíček, Kateřina Kubecová

2025-12-10

MPLS, again

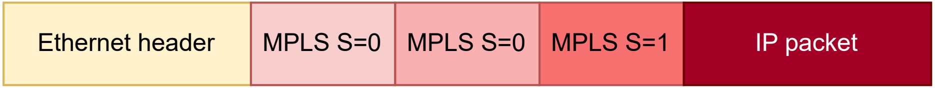
- Silicon capable of IP routing is expensive
- MPLS uses fixed-size labels
- Allows traffic flow engineering and VPNs
- Network of nodes with cheap silicon
- Stackable data

MPLS Label Stack

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | Exp |S|       TTL     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

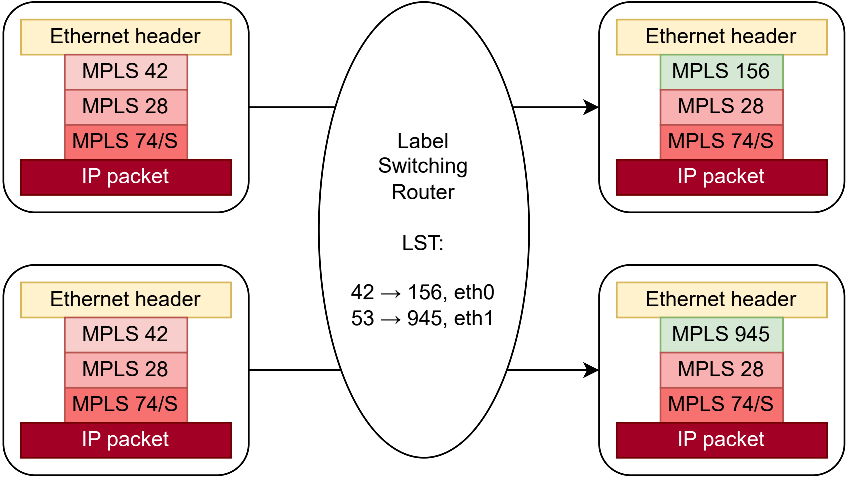
- Label: 20 bits of value
  special values: 0–15
- Exp: experimental (set to 0)
- S: bottom of stack
- TTL: time to live

MPLS Label Stack



- No explicit next header
- Must be encoded in the MPLS stack
- Special label values:
  - Explicit zero (bottom, route by IPv4): 0
  - Router Alert (anywhere but bottom): 1
  - Explicit zero (bottom, route by IPv6): 2
  - Implicit NULL (only in signalling): 3
  - Reserved: 4–15

MPLS Label Switching



Forwarding Equivalence Class

- Traffic flows which should behave the same
- Basic concepts:
  - Data forwarded to a specific edge router
  - Data sent out to a specific link
  - Data forwarded to a specific neighbor router
- More complex examples:
  - Data for a specific service (e.g. HTTP endpoints)
  - Private data of a customer (VPN)
- Intermediate routers do simple operations
- Route set by ingress up to egress
- FEC is always defined by an IGP protocol

Label Distribution Protocol

- Advertise mapping Label $\leftrightarrow$ FEC
- One FEC may have multiple labels
- One label must point to one FEC
- **Mapping is local to every node**
- Every node assigns a **local label** to FECs
- Label Switching Table:
  - Local label
  - FEC (whatever)
  - Destination (neighbor)
  - Destination-specific label

LDP on the wire

- UDP multicast Hello
- Received UDP Hello? Open a TCP session!
- Exchange FECs and label mappings
- Different modes of operation
  - Send everything outright / Send only on demand
  - Label everything outright / Label only on demand

LDP Targeted Hello

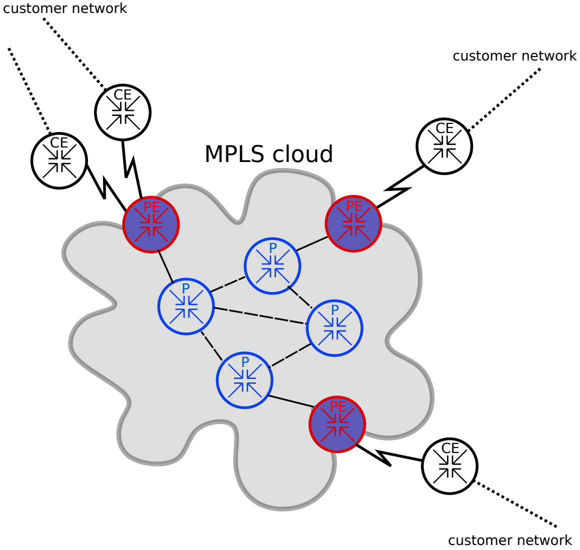
- Internal overlay
- Multihop Hello, multihop TCP connection
- Additional layer of MPLS labels
- Useful when the network is too big

BGP L3VPN Overview

- Layer 3 VPN service using BGP as control plane
- Provides isolated IP connectivity between customer sites
- Uses MPLS backbone for packet forwarding
- RFC 4364

Terminology

- PE (Provider edge) router
- CE (Customer edge) router
- P (Provider) router



PE Router

- Customer-facing router in provider network
- Maintains customer routing tables (VRFs)
- Exchanges IP routes with CEs (any protocol)
- Runs BGP to exchange VPN routes with other PEs
- Encapsulates and decapsulates VPN traffic in MPLS
- Attachment circuits to CE routers

CE and P Routers

CE (Customer edge) router

- Customer's router at site edge
- Connected to PE router (or multiple ones)
- No VPN awareness, just IP forwarding
- Can be just host device

P (Provider) Router

- Core router in provider network
- No VPN awareness, IP and MPLS forwarding

VRF (Virtual Routing and Forwarding)

- Multiple routing domains inside single router
- Per-VRF routing and forwarding table
- Set of interfaces bound to specific VRF
- Ingress traffic from bound interfaces uses per-VRF table
- Route next hops may point to another VRFs or VRF-external interfaces
- Handles overlapping IP space (private ranges)

VPN Data Path

1. CE sends packet to PE
2. PE performs lookup in VRF table
3. PE adds **two** MPLS labels and sends it to network
4. P routers forward based on outer label
5. Egress PE removes labels and selects VRF based on inner one
6. Egress PE performs lookup in selected VRF table and forwards to CE

Two-label stack: Outer (transport) + Inner (VRF)

VPN Data Path Optimizations

- Egress PE performs three lookups (outer label, inner label, IP in VRF)
- Outer label removed by last P router (penultimate hop popping)
- Inner label bound to CE next hop instead of whole VRF

VPN Routes

- Based on IP routes in VRF tables
- IP prefix is extended by route distinguisher (RD)
- Marked by route targets (RTs) as extended communities
- Stored in shared VPN table on PE
- MPLS-labeled

Route Distinguishers

- 64-bit identifier to make routes unique
- Allows overlapping IP addresses between different VPNs
- No inherent meaning, not used for routing decisions
- Typically 1 RD per VRF per PE
- Should be globally unique (?)
- Format *global-id*:*local-id*

Route Distinguishers

- Type (2 bytes), Value (6 bytes)
- Type 0: *ASN*:*num* (2-byte ASN, 4-byte number)
  Example: `65001:100000`
- Type 1: *IP*:*num* (4-byte IP, 2-byte number)
  Example: `192.168.1.1:100`
- Type 2: *ASN*:*num* (4-byte ASN, 2-byte number)
  Example: `4200000001:100`

Route Targets

- 64-bit extended communities
- Controls route import/export between VRFs
- Determines VPN membership
- Typically 1 RD per VPN (on all PEs)
- Multiple RTs on one route possible
- Format (`rt`, *global-id*, *local-id*), `target`:*global-id*:*local-id*

Route Targets

- Export target: Attached when route leaves VRF
- Import target: Determines which routes enter VRF
- Same RT: Bidirectional communication
- Different RTs: Unidirectional or hub-and-spoke topologies

Route Targets

- Type (2 bytes), Value (6 bytes)
- Type 0x0002: (`rt`, *ASN*, *num*) (2-byte ASN, 4-byte number)
  Example: `(rt, 65001, 100000)`
- Type 0x0102: (`rt`, *IP*, *num*) (4-byte IP, 2-byte number)
  Example: `(rt, 192.168.1.1, 100)`
- Type 0x0202: (`rt`, *ASN*, *num*) (4-byte ASN, 2-byte number)
  Example: `(rt, 4200000001, 100)`

BGP L3VPN

- Carries VPN routes between PEs
- Uses VPNv4 / VPNv6 address family (SAFI 128)
- NLRI format: RD + IP prefix (+ MPLS labels)
- Route targets are just extended communities

PE Router Tables

- Regular IP routing table (from OSPF/other IGP)
- LSP (label switched path) table (from LDP+OSPF / other)
- MPLS table (forwarding entries for local labels)
- VPN table (for all VPN routes)
- VRF IP tables (one per VRF)

VPN Route Propagation

1. PE learns IP route from CE and stores it to VRF table
2. IP route is converted to VPN route and stored to common VPN table
   - IP prefix extended by VRF route distinguisher
   - Route target extended communities set to VRF export targets
   - Inner MPLS label assigned based on VRF policy
3. VPN route is propagated by BGP to other PEs, IP of this PE as BGP next hop
4. On other PEs, BGP next hop is resolved through LSP table, to get immediate next hop and outer MPLS label
5. For each VRFs with matching import targets, received VPN route is converted to IP route and stored in VRF table, immediate next hop with both labels are kept

Alternative Encapsulations

- Do we really need MPLS network for this?
- MPLS in IP or GRE (RFC 4023)
- BGP Tunnel Encapsulation Attribute (RFC 9012)
- MPLS network replaced by tunnels
- ‘Inner’ MPLS label still used to distinguish VPNs