

MPLS Enlightened Open vSwitch

Netfilter Workshop, Montpellier

Simon Horman <simon@horms.net>
Horms Solutions Ltd.

July 2014

- Multi-Layer Software Switch

- Multi-Layer Software Switch
- Datapath provides fast-path for known flows
 - Two implementations: Userspace and Linux Kernel

- Multi-Layer Software Switch
- Datapath provides fast-path for known flows
 - Two implementations: Userspace and Linux Kernel
- ovs-vswitchd is the brains of the operation
 - In userspace
 - Flow setup using wild-carded Open Flow flows
 - May connect to Open Flow controller

- Protocol for control of switches
- Switch connects to controller
- Switch may send missed-packets to controller
- Controller may add flows to switch

Multi Protocol Label Switching

- A packet may include a label stack which is composed of one or more label stack entries (LSEs)
- Inserted just before L3 header

Anatomy of an MPLS LSE

- An MPLS LSE contains:
 - Label
 - Traffic Class (TC)
 - Bottom of Stack (BoS) bit
 - TTL



- match: MPLS Label, TC and BoS bit

OpenFlow and MPLS: Basic Support

- match: MPLS Label, TC and BoS bit
- set_field: MPLS Label and TC

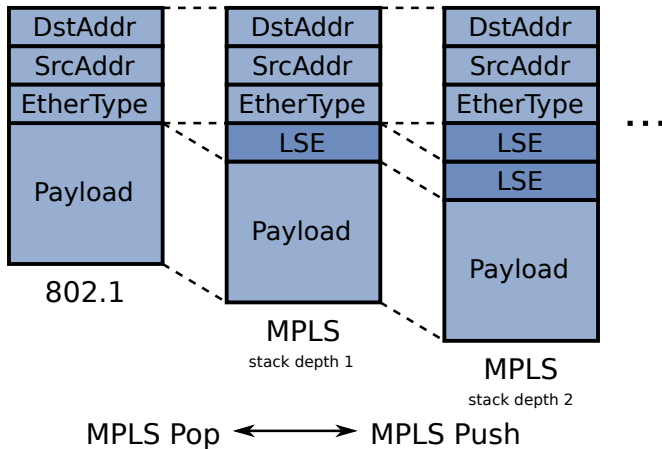
OpenFlow and MPLS: Basic Support

- match: MPLS Label, TC and BoS bit
- set_field: MPLS Label and TC
- Set MPLS TTL
- Dec MPLS TTL

OpenFlow and MPLS: Basic Support

- match: MPLS Label, TC and BoS bit
- set_field: MPLS Label and TC
- Set MPLS TTL
- Dec MPLS TTL
- MPLS push
- MPLS pop

MPLS Push and Pop



Status of Basic Support

Status:

- ovs-vsctl: Merged
- user-space datapath: Merged
- Linux kernel datapath: Merged very recently (patchset v2.62)
- Linux network core: GSO MPLS: Merged (v3.10, v3.16-rc1)

Complications Part I: Background

- L3+ actions applied to MPLS packets.
e.g.: `mpls_pop(0x0800),dec_ttl`

Complications Part I: Background

- L3+ actions applied to MPLS packets.
e.g.: `mpls_pop(0x0800),dec_ttl`
- MPLS LSE does not include EtherType



Complications Part I: Background

- L3+ actions applied to MPLS packets.
e.g.: `mpls_pop(0x0800),dec_ttl`
- MPLS LSE does not include EtherType



- Rather, it is supplied by the MPLS pop action

Complications Part I: Background

- L3+ actions applied to MPLS packets.
e.g.: `mpls_pop(0x0800),dec_ttl`
- MPLS LSE does not include EtherType



- Rather, it is supplied by the MPLS pop action
- So decoding of the inner-packet can't occur until an MPLS action is applied

Complications Part II: Circular Dependency

- The actions for a packet actions aren't known until a flow is looked-up

Complications Part II: Circular Dependency

- The actions for a packet actions aren't known until a flow is looked-up
- But the L3 portion of the match can't be filled in until an MPLS pop action is applied

- Allow a packet to be re-injected after actions are applied

Solution: Recirculation

- Allow a packet to be re-injected after actions are applied
e.g.:
 - 1 Lookup flow
 - 2 Apply actions, modifying packet
 - 3 Lookup flow for modified packet
 - 4 Apply actions

Solution: Recirculation

- Allow a packet to be re-injected after actions are applied
e.g.:
 - 1 Lookup flow
 - 2 Apply actions, modifying packet
 - 3 Lookup flow for modified packet
 - 4 Apply actions
- Status: Merged last month (2nd rewrite)

Questions