Configuring MPLS

The basic MPLS actions are Push, Swap and Pop. Beginning with PicOS 2.4, user does not need to set the ***eth_src*** in actions during Push MPLS. The packet **src_mac** pushes the MPLS to the MAC address of this switch.

User can add flows to modify and copy the MPLS TTL and IP TTL.

User can push/pop 2 MPLS labels per flow.

> Every un-tagged packet is tagged with the default VLAN-ID before Push, Pop and Swap.

Hardware or Software based Forwarding

The flow is pre-installed in hardware if there is enough information on the Flow to be processed by the ASIC.

Here is the minimal set of information required in flow match to process the packet on hardware only,**before version 2.8.1**:

If the Flow action is a POP : dl_dst, dl_vlan, mpls_lse
If the Flow action is a PUSH : dl_dst, dl_vlan, dl_type,mpls_lse (only being needed when dl_type is 0x8847 or 0x8848)
If the Flow action is a SWAP :dl_dst, dl_vlan, mpls_lse

Since version 2.8.1,configuration is changed,the minimal set on hardware only as shown below:

If the Flow action is a PUSH:
push mpls for ip packet: in_port,dl_vlan,dl_dst,dl_type
push mpls for mpls packet: in_port,dl_vlan,dl_dst,dl_type(mpls),mpls_label

If the Flow action is a SWAP: in_port,dl_vlan,dl_dst,dl_type(mpls),mpls_label

If the Flow action is a POP: in_port,dl_vlan,dl_dst,dl_type(mpls),mpls_label

> 1. There is one exception. If the action is "pop_mpls:0x8847" and matches enough fields (dl_dst,vlan_id,mpls_label), the flow becomes a direct flow (hardware only).
>
> 2. If the flow actions a push, flow match must include dl_type=0x0800/dl_type=0x86dd/dl_type=0x8847/dl_type=0x8848. If the flow match not include one of them, the flow entry cannot add.

If there is some information missing to process the packet, the flow becomes a "packet-driven" flow. This means that the first MPLS packet is sent to the CPU, which analyzes it and downloads a new flow on the hardware with the missing information to handle the packet in hardware. Following packets for this specific flow will then be handled by the hardware (ASIC) without reaching the Switch CPU.

PUSH MPLS

Pushing an MPLS Label

In the following configuration, user specifies a flow, which should match:
{ in_port=1,dl_type=0x0800, dl_dst=22:00:00:00:00:00,dl_vlan=1}
The action is to push an MPLS label (i.e. 10) and forward to port te-1/1/2
Note that MPLS TTL will copy from the IP header and decrease

Pushing two MPLS Labels

```
root@PicOS-OVS# ovs-ofctl add-flow br0 in_port=1,dl_type=0x0800,dl_dst=22:00:00:00:00:00,dl_vlan=1,
actions=push_mpls:0x8847,set_field:10-\>mpls_label,output:2
```

In the following configuration, specify a flow, which should match { in_port=1,dl_type=0x0800,dl_dst=22:00:00:00:00:00,dl_vlan=1}, the action is to push two labels (i.e. 10 and 20) /EXP/TTL and forward to port te-1/1/2

```
root@PicOS-OVS# ovs-ofctl add-flow br0 in_port=1,dl_type=0x0800,dl_dst=22:00:00:00:00:00,dl_vlan=1,
actions=push_mpls:0x8847,set_field:10-\>mpls_label,push_mpls:0x8847,set_field:20-\>mpls_label,set_field:2-
>mpls_tc,set_mpls_ttl:30,output:2
root@PicOS-OVS#
```

SWAP MPLS Label

Swapping MPLS Labels

In following configuration, user specifies a flow, which should match { in_port=1,dl_type=0x8847,dl_dst=22:00:00:00:00:00,dl_vlan=1,mpls_label=10}, the action is to swap label 10 with 20 and forward to port te-1/1/2

```
root@PicOS-OVS# ovs-ofctl add-flow br0 in_port=1,dl_type=0x8847,dl_dst=22:00:00:00:00:00,dl_vlan=1,
mpls_label=10,actions=set_field:20-\>mpls_label,output:2
root@PicOS-OVS#
```

POP MPLS Label

Popping an MPLS Label of the flow

In following configuration, specify a flow, which should match { in_port=1,dl_type=0x8847,dl_dst=22:00:00:00:00:00,dl_vlan=1,mpls_label=10}, the action is to pop MPLS label and forward to port te-1/1/2.
Note that MPLS TTL will be copied to IP header TTL and decremented by 1.

```
root@PicOS-OVS# ovs-ofctl add-flow br0 in_port=1,dl_type=0x8847,dl_dst=22:00:00:00:00:00,dl_vlan=1,
mpls_label=10,actions=pop_mpls:0x0800,output:2
```

Popping one MPLS Label for flows with Two MPLS Labels

In the following configuration, specify a flow that has two MPLS labels (i.e. 10 and 20). The pop action is always popping the outer MPLS header. Note that the two label flow pops only one label, the output packet is also a MPLS packet. Thus, the "pop_mpls:0x8847" must be configured.

```
root@PicOS-OVS# ovs-ofctl add-flow br0 in_port=1,dl_type=0x8847,dl_dst=22:00:00:00:00:00,dl_vlan=1,
mpls_label=10,mpls_label2=20,actions=pop_mpls:0x8847,output:2
```

Popping two MPLS Labels for flows with two MPLS Lables

In following configuration, specify a flow which has two labels to pop. The output flow is IP packet. Configure two pop entries to pop the flow.

```
root@PicOS-OVS# ovs-ofctl add-flow br0 in_port=1,dl_type=0x8847,mpls_label=10,mpls_label2=20,dl_dst=22:00:00:00:
00:00,dl_vlan=1,actions=pop_mpls:0x0800,output:2
```

notice :

if the action is pop_mpls:0x0800 to pop two mpls label, must have to match two mpls label.

PUSH MPLS Label and VLAN

Pushing one MPLS Label and one VLAN

In following configuration, specify flows to push one MPLS Label and one VLAN

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x0800,dl_vlan=2999,dl_dst=22:22:22:22:22:22,actions=push_mpls:0x8847,
set_field:333-\>mpls_label,push_vlan:0x8100,set_field:1999-\>vlan_vid,output:4
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,dl_dst=22:22:22:22:22:22,mpls_label=66,
actions=push_mpls:0x8847,set_field:333-\>mpls_label,output:4
```

Pushing two MPLS Labels and one VLAN

In following configuration, specify flows to push two mpls labels and one VLAN

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x0800,dl_vlan=2999,dl_dst=22:22:22:22:22:22,actions=push_mpls:0x8847,
set_field:333-\>mpls_label,push_mpls:0x8847,set_field:66-\>mpls_label,push_vlan:0x8100,set_field:1999-
\>vlan_vid,output:4
```

POP One or Two MPLS Labels

In following configuration, specify flows which should match dl_type,dl_vlan and the MPLS label. The action is to pop one MPLS label and set a new src mac address. The first flow will pop one MPLS label and the second flow will pop two MPLS labels.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,mpls_label=111,dl_dst=22:22:22:22:
22:22,actions=pop_mpls:0x8847,set_field:22:22:22:44:44:44-\>eth_src,output:4
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,mpls_label=111,actions=pop_mpls:
0x0800,set_field:22:22:22:44:44:44-\>eth_src,output:4
```

POP One or Two MPLS Labels and PUSH/SWITCH/POP VLAN

The following flow should match dl_type and dl_vlan, action is to pop one MPLS label and push one VLAN.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,mpls_label=111,dl_dst=22:22:22:22:
22:22,actions=pop_mpls:0x8847,push_vlan:0x8100,set_field:1999-\>vlan_vid,output:4
```

The following flow should match dl_type and dl_vlan, with the action of popping two MPLS labels and pushing one VLAN.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,mpls_label=111,dl_dst=22:22:22:22:
22:22,actions=pop_mpls:0x0800,push_vlan:0x8100,set_field:1999-\>vlan_vid,output:4
```

The following flow should match dl_type and dl_vlan and mpls label, and the action is to pop one MPLS labels and swap VLAN.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,dl_dst=22:22:22:22:22:22,
actions=pop_mpls:0x0800,push_vlan:0x8100,set_field:1999-\>vlan_vid,output:4
```

The following flow should match dl_type, the action is to pop two MPLS labels and pop VLAN.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,dl_dst=22:22:22:22:22:22,
actions=pop_mpls:0x0800,pop_vlan,output:4
```

PUSH MPLS and POP VLAN

In the following configuration, push MPLS label and pop VLAN have been supported. Following flow, the action is to push one MPLS label and pop one VLAN.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,dl_dst=22:22:22:22:22:22,
actions=push_mpls:0x8847,set_field:333-\>mpls_label,pop_vlan,output:4
```

SWITCH MPLS Labels and MOD/POP VLAN

Following flow, action is to push two MPLS labels and pop VLAN.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,dl_dst=22:22:22:22:22:22,
actions=set_field:111-\>mpls_label,pop_vlan,output:4
```

Following flow, action is to push two MPLS labels and mod VLAN.

```
ovs-ofctl add-flow br0 in_port=2,dl_type=0x8847,dl_vlan=2999,mpls_label=333,dl_dst=22:22:22:22:22:22,
actions=set_field:111-\>mpls_label,mod_vlan_vid:11,output:4
```

NOTICE:

1. If flows match dl_vlan and the actions have push_vlan, then receives packets only carry the pushed vlan.

2. Hardware can’t support pop_mpls and pop_vlan at the same time, and the packets can’t forward with line-speed.

3. When actions don’t appoint to modify dl_src, the src mac of received packets should be modified to bridge mac whatever for direct flows or packet-driven flow.

4. Push two mpls labels is supported, but push two vlans at the same time is not supported.

5. Hardware can't support push two vlan_header.

Match or Pop Reserved Mpls Label 0/2/3

From version 2.9.2, mpls support to match or pop reserved mpls label.

A value of 0 represents the "IPv4 Explicit NULL Label",this label value is only legal at the bottom of the label stack.

A value of 1 represents the "Router Alert Label", this label value is legal anywhere in the label stack except at the bottom.

A value of 2 represents the "IPv6 Explicit NULL Label",this label value is only legal at the bottom of the label stack.

A value of 3 represents the "Implicit NULL Label",this is a label that an LSR may assign and distribute, but which never actually appears in the encapsulation.

Example 1,match mpls label=0 and then pop_mpls:0x8847.

```
admin@PICOS-OVS$ovs-ofctl add-flow br0 in_port=1,mpls,mpls_label=0,dl_dst=22:22:22:22:22:22,dl_vlan=199,
actions=pop_mpls:0x8847,output:14
```

Send ipv4 packets with mpls_label=0 to match the flow, te-1/1/14 receive packets without mpls label.

Example 2,match mpls label2=0 and then pop_mpls:0x8847.

```
admin@PICOS-OVS$ovs-ofctl add-flow br0 in_port=1,mpls,mpls_label=100,mpls_label2=0,dl_dst=22:22:22:22:22:22,
dl_vlan=199,actions=pop_mpls:0x8847,output:14
```

Send ipv4 packets with mpls_label=100 and mpls_label2=0 to match the flow, te-1/1/14 receive packets with mpls_label=0.

Example 3,match mpls label=2 and then pop_mpls:0x8847.

```
admin@PICOS-OVS$ovs-ofctl add-flow br0 in_port=1,mpls,mpls_label=2,dl_dst=22:22:22:22:22:22,dl_vlan=199,
actions=pop_mpls:0x0800,output:14
```

Send ipv6 packets with mpls_label=2 to match the flow, te-1/1/14 receive packets without mpls label.

Example 4,match mpls label2=2 and then pop_mpls:0x0800.

```
admin@PICOS-OVS$ovs-ofctl add-flow br0 in_port=1,mpls,mpls_label=100,mpls_label2=2,dl_dst=22:22:22:22:22:22,
dl_vlan=199,actions=pop_mpls:0x0800,output:14
```

Send ipv6 packets with mpls_label=100 and mpls_label2=2 to match the flow, te-1/1/14 receive packets without mpls label.

⚠ 1.It's not allowed to match/pop label 3. Additionally a label of 3 should not be included in MPLS encapsulation.
2.Platforms which support mpls:

3296 3780 3920 3922 3924 3930 5101 5401
* arctica4806xp as5712_54x as5812 as6701_32x as6712_32x as7712_32x
* dcs7032q28 es4654bf niagara2632xl niagara2948-6xl s4048 z9100 msh8920

2.Platforms which not support mpls:

3290 3295 as4610