

NAME

mpls - mpls manipulation module

SYNOPSIS

tc ... action mpls { *POP* | *PUSH* | *MODIFY* | **dec_ttl** } [*CONTROL*]

POP := **pop** **protocol** *MPLS_PROTO*

PUSH := { **push** | **mac_push** } [**protocol** *MPLS_PROTO*] [**tc** *MPLS_TC*] [**ttl** *MPLS_TTL*] [**bos** *MPLS_BOS*] **label** *MPLS_LABEL*

MODIFY := **modify** [**label** *MPLS_LABEL*] [**tc** *MPLS_TC*] [**ttl** *MPLS_TTL*]

CONTROL := { **reclassify** | **pipe** | **drop** | **continue** | **pass** | **goto chain** *CHAIN_INDEX* }

DESCRIPTION

The **mpls** action performs mpls encapsulation or decapsulation on a packet, reflected by the operation modes *POP*, *PUSH*, *MODIFY* and *DEC_TTL*. The *POP* mode requires the ethertype of the header that follows the MPLS header (e.g. IPv4 or another MPLS). It will remove the outer MPLS header and replace the ethertype in the MAC header with that passed. The *PUSH* and *MODIFY* modes update the current MPLS header information or add a new header. *PUSH* requires at least an *MPLS_LABEL*. *DEC_TTL* requires no arguments and simply subtracts 1 from the MPLS header TTL field.

OPTIONS

pop Decapsulation mode. Requires the protocol of the next header.

push Encapsulation mode. Adds the MPLS header between the MAC and the network headers. Requires at least the **label** option.

mac_push

Encapsulation mode. Adds the MPLS header before the MAC header. Requires at least the **label** option.

modify Replace mode. Existing MPLS tag is replaced. **label**, **tc**, and **ttl** are all optional.

dec_ttl Decrement the TTL field on the outer most MPLS header.

label *MPLS_LABEL*

Specify the MPLS LABEL for the outer MPLS header. *MPLS_LABEL* is an unsigned 20bit integer, the format is detected automatically (e.g. prefix with '0x' for hexadecimal interpretation, etc.).

protocol *MPLS_PROTO*

Choose the protocol to use. For push actions this must be **mpls_uc** or **mpls_mc** (**mpls_uc** is the default). For pop actions it should be the protocol of the next header. This option cannot be used with modify.

tc *MPLS_TC*

Choose the TC value for the outer MPLS header. Decimal number in range of 0-7. Defaults to 0.

ttl *MPLS_TTL*

Choose the TTL value for the outer MPLS header. Number in range of 0-255. A non-zero default value will be selected if this is not explicitly set.

bos *MPLS_BOS*

Manually configure the bottom of stack bit for an MPLS header push. The default is for TC to automatically set (or unset) the bit based on the next header of the packet.

CONTROL

How to continue after executing this action.

reclassify

Restarts classification by jumping back to the first filter attached to this action's parent.

pipe

Continue with the next action, this is the default.

drop

Packet will be dropped without running further actions.

continue

Continue classification with next filter in line.

pass

Return to calling qdisc for packet processing. This ends the classification process.

EXAMPLES

The following example encapsulates incoming IP packets on eth0 into MPLS with a label 123 and sends them out eth1:

```
#tc qdisc add dev eth0 handle ffff: ingress
#tc filter add dev eth0 protocol ip parent ffff: flower \
    action mpls push protocol mpls_uc label 123 \
    action mirred egress redirect dev eth1
```

In this example, incoming MPLS unicast packets on eth0 are decapsulated and redirected to eth1:

```
#tc qdisc add dev eth0 handle ffff: ingress
#tc filter add dev eth0 protocol mpls_uc parent ffff: flower \
    action mpls pop protocol ipv4 \
    action mirred egress redirect dev eth1
```

Here is another example, where incoming Ethernet frames are encapsulated into MPLS with label 123 and TTL 64. Then, an outer Ethernet header is added and the resulting frame is finally sent on eth1:

```
#tc qdisc add dev eth0 ingress
#tc filter add dev eth0 ingress matchall \
    action mpls mac_push label 123 ttl 64 \
    action vlan push_eth \
        dst_mac 02:00:00:00:00:02 \
        src_mac 02:00:00:00:00:01 \
    action mirred egress redirect dev eth1
```

The following example assumes that incoming MPLS packets with label 123 transport Ethernet frames. The outer Ethernet and the MPLS headers are stripped, then the inner Ethernet frame is sent on eth1:

```
#tc qdisc add dev eth0 ingress
#tc filter add dev eth0 ingress protocol mpls_uc \
    flower mpls_label 123 mpls_bos 1 \
    action vlan pop_eth \
    action mpls pop protocol teb \
    action mirred egress redirect dev eth1
```

SEE ALSO

tc(8), **tc-mirred(8)**, **tc-vlan(8)**