

NAME

skbmod - user-friendly packet editor action

SYNOPSIS

```
tc ... action skbmod { set SETTABLE | swap SWAPPABLE | ecn } [ CONTROL ] [ index INDEX ]
```

```
SETTABLE := [ dmac DMAC ] [ smac SMAC ] [ etype ETYPE ]
```

```
SWAPPABLE := mac
```

```
CONTROL := { reclassify | pipe | drop | shot | continue | pass }
```

DESCRIPTION

The **skbmod** action is intended as a usability upgrade to the existing **pedit** action. Instead of having to manually edit 8-, 16-, or 32-bit chunks of an ethernet header, **skbmod** allows complete substitution of supported elements. Action must be one of **set**, **swap** and **ecn**. **set** and **swap** only affect Ethernet packets, while **ecn** only affects IP packets.

OPTIONS

dmac *DMAC*

Change the destination mac to the specified address.

smac *SMAC*

Change the source mac to the specified address.

etype *ETYPE*

Change the ethertype to the specified value.

mac Used to swap mac addresses.

ecn Used to mark ECN Capable Transport (ECT) IP packets as Congestion Encountered (CE). Does not affect Non ECN-Capable Transport (Non-ECT) packets.

CONTROL

The following keywords allow to control how the tree of qdisc, classes, filters and actions is further traversed after this action.

reclassify

Restart with the first filter in the current list.

pipe

Continue with the next action attached to the same filter.

drop

shot Drop the packet.

continue

Continue classification with the next filter in line.

pass

Finish classification process and return to calling qdisc for further packet processing. This is the default.

EXAMPLES

To start, observe the following filter with a pedit action:

```
tc filter add dev eth1 parent 1: protocol ip prio 10 \
    u32 match ip protocol 1 0xff flowid 1:2 \
    action pedit munge offset -14 u8 set 0x02 \
    munge offset -13 u8 set 0x15 \
    munge offset -12 u8 set 0x15 \
    munge offset -11 u8 set 0x15 \
    munge offset -10 u16 set 0x1515 \
    pipe
```

Using the skbmod action, this command can be simplified to:

```
tc filter add dev eth1 parent 1: protocol ip prio 10 \  
    u32 match ip protocol 1 0xff flowid 1:2 \  
    action skbmod set dmac 02:15:15:15:15:15 \  
    pipe
```

Complexity will increase if source mac and ethertype are also being edited as part of the action. If all three fields are to be changed with skbmod:

```
tc filter add dev eth5 parent 1: protocol ip prio 10 \  
    u32 match ip protocol 1 0xff flowid 1:2 \  
    action skbmod \  
    set etype 0xBEEF \  
    set dmac 02:12:13:14:15:16 \  
    set smac 02:22:23:24:25:26
```

To swap the destination and source mac addresses in the Ethernet header:

```
tc filter add dev eth3 parent 1: protocol ip prio 10 \  
    u32 match ip protocol 1 0xff flowid 1:2 \  
    action skbmod \  
    swap mac
```

Finally, to mark the CE codepoint in the IP header for ECN Capable Transport (ECT) packets:

```
tc filter add dev eth0 parent 1: protocol ip prio 10 \  
    u32 match ip protocol 1 0xff flowid 1:2 \  
    action skbmod \  
    ecn
```

Only one of **set**, **swap** and **ecn** shall be used in a single command. Trying to use more than one of them in a single command is considered undefined behavior; pipe multiple commands together instead.

SEE ALSO

tc(8), **tc-u32(8)**, **tc-pedit(8)**