

**NAME**

vlan - vlan manipulation module

**SYNOPSIS**

**tc ... action vlan** { **pop** | *PUSH* | *MODIFY* } [ *CONTROL* ]

*PUSH* := **push** [ **protocol** *VLANPROTO* ] [ **priority** *VLANPRIO* ] **id** *VLANID*

*MODIFY* := **modify** [ **protocol** *VLANPROTO* ] [ **priority** *VLANPRIO* ] **id** *VLANID*

*CONTROL* := { **reclassify** | **pipe** | **drop** | **continue** | **pass** | **goto chain** *CHAIN\_INDEX* }

**DESCRIPTION**

The **vlan** action allows to perform 802.1Q en- or decapsulation on a packet, reflected by the operation modes *POP*, *PUSH* and *MODIFY*. The *POP* mode is simple, as no further information is required to just drop the outer-most VLAN encapsulation. The *PUSH* and *MODIFY* modes require at least a *VLANID* and allow to optionally choose the *VLANPROTO* to use.

**OPTIONS**

**pop** Decapsulation mode, no further arguments allowed.

**push** Encapsulation mode. Requires at least **id** option.

**modify** Replace mode. Existing 802.1Q tag is replaced. Requires at least **id** option.

**id** *VLANID*

Specify the VLAN ID to encapsulate into. *VLANID* is an unsigned 16bit integer, the format is detected automatically (e.g. prefix with '0x' for hexadecimal interpretation, etc.).

**protocol** *VLANPROTO*

Choose the VLAN protocol to use. At the time of writing, the kernel accepts only **802.1Q** or **802.1ad**.

**priority** *VLANPRIO*

Choose the VLAN priority to use. Decimal number in range of 0-7.

*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 ICMP packets on eth0 from 10.0.0.2 into VLAN ID 123:

```
#tc qdisc add dev eth0 handle ffff: ingress
#tc filter add dev eth0 parent ffff: pref 11 protocol ip \
    u32 match ip protocol 1 0xff flowid 1:1 \
        match ip src 10.0.0.2 flowid 1:1 \
        action vlan push id 123
```

Here is an example of the **pop** function: Incoming VLAN packets on eth0 are decapsulated and the classification process then restarted for the plain packet:

```
#tc qdisc add dev eth0 handle ffff: ingress
#tc filter add dev $ETH parent ffff: pref 1 protocol 802.1Q \
    u32 match u32 0 0 flowid 1:1 \
    action vlan pop reclassify
```

**SEE ALSO****tc(8)**