Improving TC Filters insertion rate

Guy Shattah, Rony Efraim

NetDev 2.2 (2017) – The Technical Conference on Linux Networking

Mellanox TECHNOLOGIES

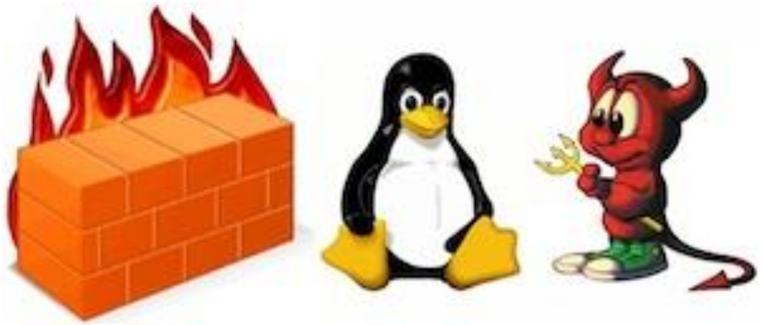
Connect. Accelerate. Outperform.®

Contents

- Connection tracking (conntrack)
- OVS CT (connection tracking)
- Current TC support
- Motivation
- Suggestion for TC
- Command line examples

Connection tracking (**conntrack)**

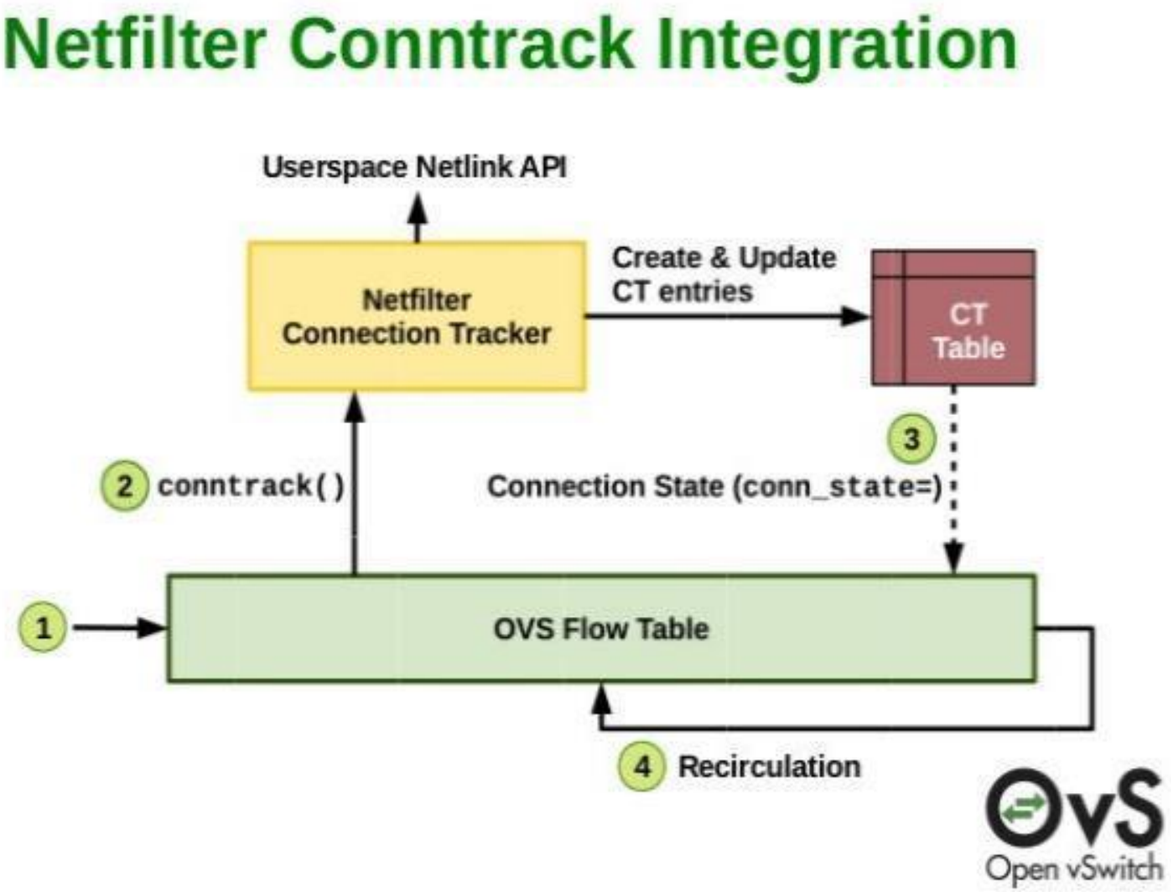
- Tracks connections and stores information about the state of connections.
- For each packet
  - Finds the connection in DB or creates a new entry.
  - Validates the packets.
    - TCP – validates that the packets are within the current TCP window and updates the window according to the ACKs



- CT state for every packet can be
  - New – The connection is starting (SYN for TCP)
  - Established – The connection has already been established
  - Related - The connection is related to an establish connection.
  - Invalid - packets do not follow the expected behavior of a connection

OVS CT (connection tracking)

- **OVS CT using the same Conntrack of the IpTable.**
- **There is an OVS action to go to the CT**
- **After CT it continues the steering with the CT state:**
  New, established, related ,reply or invalid



OVS CT (connection tracking) Example

Example of OVD datapath kernel rules :

[root@dev-r-vrt-234-005 ~]# ovs-dpctl dump-flows

recirc_id(0),in_port(5),ct_state(-trk),eth_type(0x0800),ipv4(frag=no), packets:4, bytes:300, used:2.230s, flags:P., actions:ct,recirc(0x9)

recirc_id(0xa),in_port(6),ct_state(+est+trk),eth_type(0x0800),ipv4(frag=no), packets:4, bytes:468, used:2.230s, flags:P., actions:5

recirc_id(0x9),in_port(5),ct_state(+est+trk),eth_type(0x0800),ipv4(frag=no), packets:4, bytes:300, used:2.230s, flags:P., actions:6

recirc_id(0),in_port(6),ct_state(-trk),eth_type(0x0800),ipv4(frag=no), packets:4, bytes:468, used:2.231s, flags:P., actions:ct,recirc(0xa)

Current TC support

- connmark action
  - tc ... action connmark [zone u16_zone_index] [CONTROL] [index u32_index]
  - Used to restore the connection's mark value into the packet's fwmark
  - Can't be used to classify on CT info ☹

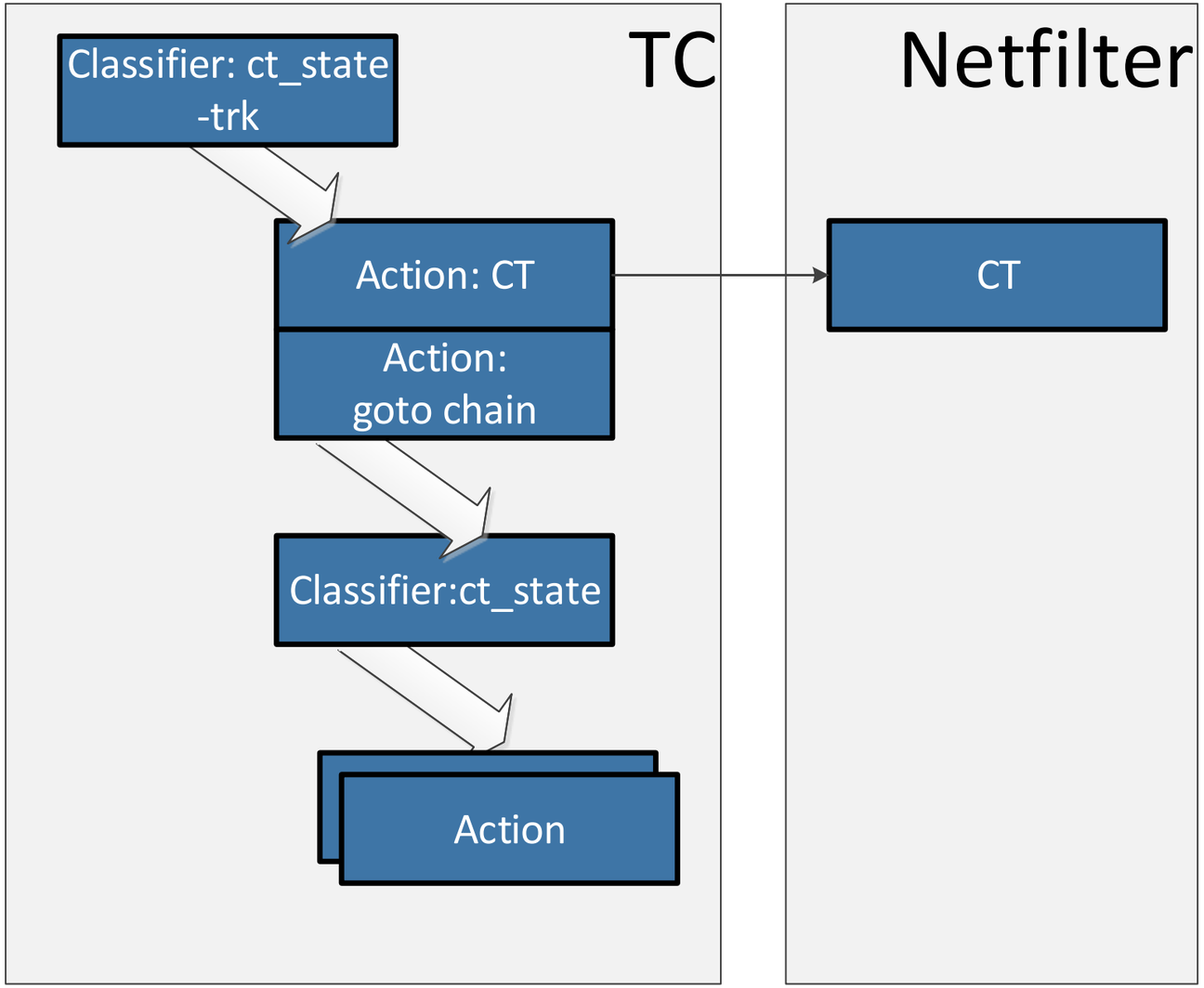
- Multi-table/Multi-chain – (= **recirc_id)**
  - Action “goto chain CHAIN_INDEX” used to go to different table
    - $ tc filter add dev eth1 parent ffff: protocol ip pref 10 flower src_ip 192.168.101.1 action **goto chain 100**
  - Classification chain CHAIN_INDEX used to add rule to specific table.
    - $ tc filter add dev eth1 parent ffff: protocol ip **chain 100** pref 10 flower dst_ip 192.168.101.1 action drop

Motivation

- Motivation of integrating Connection tracking as part of TC
  - Make TC support filters & action according to connection state.
    - OVS and iptables already support CT
    - TC has to be aligned
  - Enable OVS HW offload (skip_sw) of connection tracking.
    - Currently OVS does HW offload using TC
    - In order to utilize HW offload of CT it is required to add CT to TC

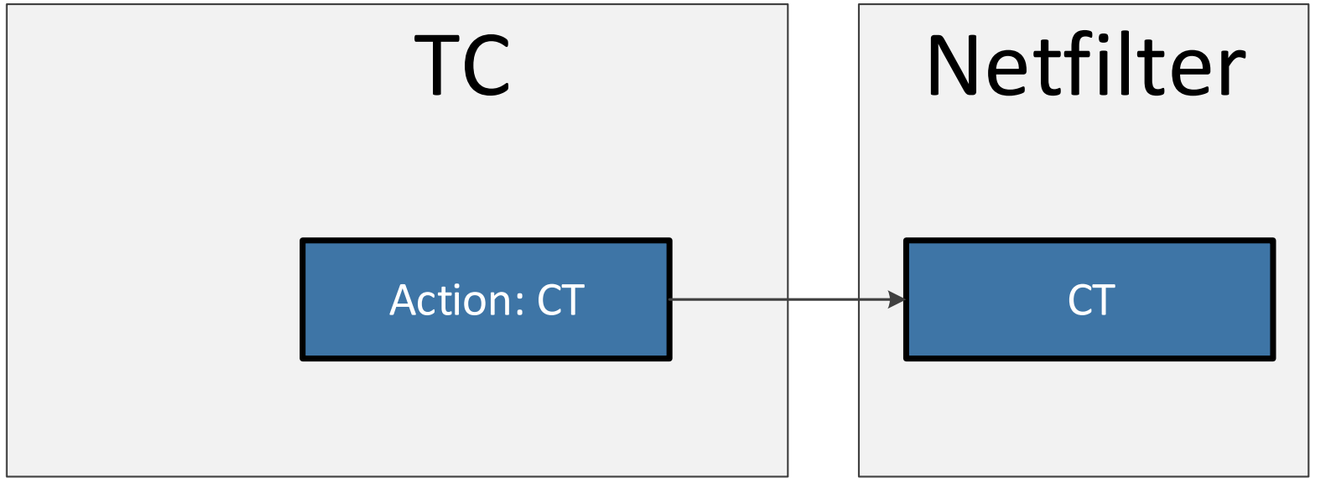
Suggestion for TC

- New action “ct” will be added in order to forward the packet to nf_ct.
- New match “ct_state” will be added to flower classifier, to classify the connection state.



Suggestion for TC

- New action “ct” will be added in order to call the nf_ct.
- The new ct action has the following optional parameters:
  - Commit
    - Commit the connection
  - Zone <number>
    - Zone number in CT to use (u16)



Suggestion for TC

- New match will be added to flower classifier call “ct_state”, to classify using the connection state.
- ct_state flags should be either set or clear
  - Set by using “+”
  - Clear by using “-“
  - All other modifiers will be ignored.
- The flags are:
  - trk - Tracked - Been through the connection tracker
  - inv - Invalid
  - new - New connection
  - est - Established connection
  - rpl - Packet is in reply direction
  - rel - Related - ICMP, eg “dst_unreach” response or helper “related” connection

Command line examples

```
tc filter add dev eth5 protocol ip parent ffff: chain 0 flower ct_state -trk
   action ct
   action goto chain 1

tc filter add dev eth6 protocol ip parent ffff: chain 0 flower ct_state -trk
   action ct
   action goto chain 2

tc filter add dev eth5 protocol ip parent ffff: chain 1 flower ct_state +trk,+est
   action mirred egress redirect dev eth6

tc filter add dev eth6 protocol ip parent ffff: chain 2 flower ct_state +trk,+est
   action mirred egress redirect dev eth5
```

Thank You

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.®