netfilter hw offloads: flow offload API

Pablo Neira Ayuso <pablo@netfilter.org

ethtool_rx and tc offloads

- Duplicated code for each subsystem:
  - ethtool_rx: layer 2, 3, 4 tuple matching (basic) + accept/drop + WoL + queue to cpu (rss ctx, vf).
    Binary blob between kernel and userspace.
  - Tc: layer 2,3,4 tuple matching + accept/drop/goto + redirect + packet edition + tunnel + checksum + mark + ratelimit (police) + sampling
    Netlink message between kernel + userspace.

Flow Rule API

- tc supports for hardware offloads:
  - Rule match: flow dissector (net/core/flow_dissector.c)
    - net/sched/cls_flower.c uses native representation
  - Rule action: tc action API
    - net/sched/act_api.c
- Add flow rule API (include/net/flow_offload.h>

```
flow_rule {
        flow_match (flow dissector)
        flow_action (based on tc action API)
}
```

- Adapt drivers to use it.

Flow block API

- Drivers set up a “flow block” via ndo_setup_tc
  - FLOW_BLOCK_SETUP type
    - FLOW_BLOCK_BIND → attach to tc block / nft basechain
    - FLOW_BLOCK_UNBIND → detach to tc block / nft basechain
  - FLOW_CLS_{REPLACE,DESTROY} type to add/delete rules
- Move tcf_block_cb to flow_block_cb in net/core/flow_offload.c

Drivers using flow offload API

- bnxt, bcm_sf2 switch
- mlx5, spectrum switch
- Nfp
- Qede
- Ocelot
- cxgb4

nf_tables_offload

- Offload flag for base chain:
  - Step 1, preparation phase → build flow rule object from nft_rule
  - Step 2, commit phase → iterate over transaction objects and call ndo_setup_offload with FLOW_CLS_SETUP (pass flow rule object)
  - Step 3, driver fills up hardware intermediate representation and configures offload.