NETRONOME

netconf: BPF progress and plans

Jakub Kicinski

Boston, May 2018

bpftool

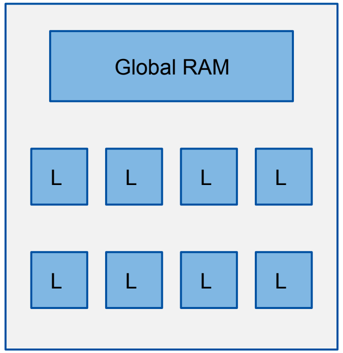
- bpffs/pin pathname search
- offload info and device-specific disassembly
- better map and helper correlation in xlated dumps
- multi-function dump support
- cgroup attach/detach
- simple BPF program load
- control flow graph of programs
- better batch support
- PERF_COUNT_SW_BPF_OUTPUT reader
- listing tracepoint/kprobe/uprobe attachment points
- 12 authors; 6 companies
- packaging (Fedora 27, RHEL)

BPF Offload (nfp)

- map offload support (hash and array maps)
- map lookup, update and delete calls from the datapath
- memcpy optimizations
- bpftool + binutils nfp disassembly
- atomic add operation (32 and 64bit)
- bpf_get_prandom_u32() support
- perf event output support
- initiate work on register state tracking (control flow and data flow analysis)
- queue selection (programmable RSS)
- specifying ifindex for program/map load in libbpf
- more instructions
- bpfilter offload? :)

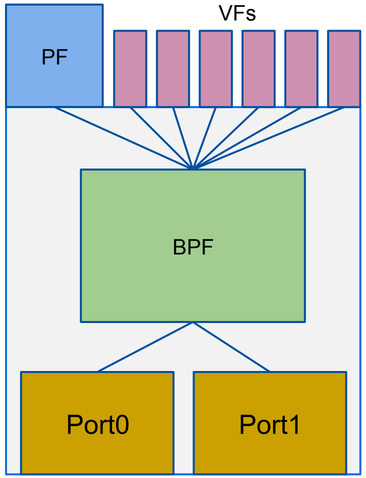
BPF Offload – Next Steps

- continue work on data flow in verfier
- multi-function programs (pseudo-calls)
- bpf_ktime_get_ns()
- xdp_adjust_tail()
- xdp_adjust_meta()
- driver XDP_REDIRECT and AF_XDP support
- simultaneous driver and offload XDP
- XDP_REDIRECT support
- XDP access to RSS hash (kernel and offload)
- per-ASIC program sharing
- distributed/read-only maps (current BPF_F_RDONLY is backwards)
- optimization/feature enable/disable
- iproute tooling does not include extack

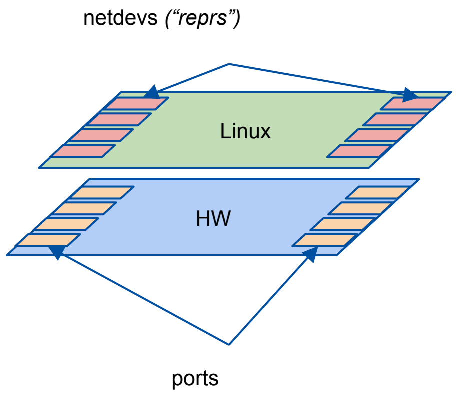


BPF-based SR-IOV e-switch

- full switchdev mode
- only legacy NDO needed is ndo_set_vf_mac
- XDP ingress on all reprs (just link TC forwarding)
- XDP_REDIRECT support
- fallback path driver XDP? AF_XDP? up to users
- per-ASIC program sharing
- ingress device from xdp_rxq_info
- dealing with mcast/bcast requires a new helper



switchdev in SR-IOV NICs



“Bridging the Gap”

NETRONOME

- enabled with a devlink switch
- today switchdev in NICs mostly used for OvS/TC flower (4 vendors)
- switchdev mode rules:
  - unless bridged/rules present all traffic to representors
  - no working starting config
  - maybe doesn’t matter in real life
- how to model a legacy SR-IOV device:
  - bridge with flooding and learning off
  - extra filtering for spoof-check
  - one bridge per uplink (or bonded uplinks)
- reuse of legacy NDOs:
  - synchronize the values?
  - reject legacy NDOs?
  - set MAC based on static FDB entries?

legacy NDO	switchdev/bridge action
link state	reprs up/down
set MAC	update FDB; update filter
set VLAN	update VLANs
spoofchk	remove MAC filter
trusted	remove filters; flooding on

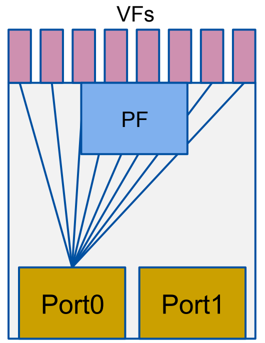
“switchdev mode” of NICs

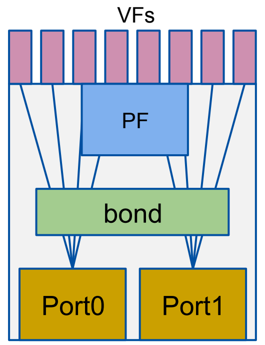
NETRONOME

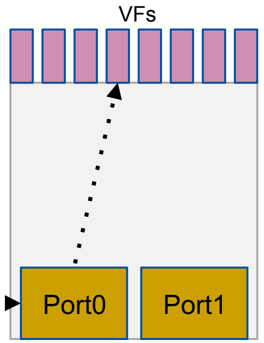
- reuse of networking APIs:
  - TC cls offloads (flower etc.)
  - bridge and port statistics for simple e-switch case
  - QoS settings with TC
  - XDP “egress” in the NIC?
  - repr netdev state as VF policy (link, max MTU, RSS?)?
  - ethtool channels for max queues?
- non-networking settings:
  - queue allocation + RSS config
  - IRQ/MSI-X allocation
  - VF allocation
  - new devlink API with persistence?
- problems:
  - not really an extension of existing SR-IOV model
  - first vendor problem
  - doesn’t work for HW which can’t do representors
  - distributed control (multi-host NICs):
    - read-only configurations
    - local-only configurations

Multi-Port NIC SR-IOV Uplink Selection

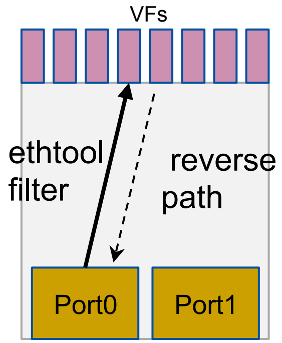
- most NICs have one PCI PF per uplink port
- PCI VFs are not associated with uplink, no 1:1 mapping
- mlx4 and nfp share the PCI PF across all ports
- need to set and get VF<>uplink association:
  - mlx4 uses num_vfs module parameter
  - static?
  - reuse the module parameter?
  - *last uplink netdev to touch the settings hack?*
  - *ethtool filter on uplink hack?*
  - new legacy NDO?
  - switchdev mode -> bridge?



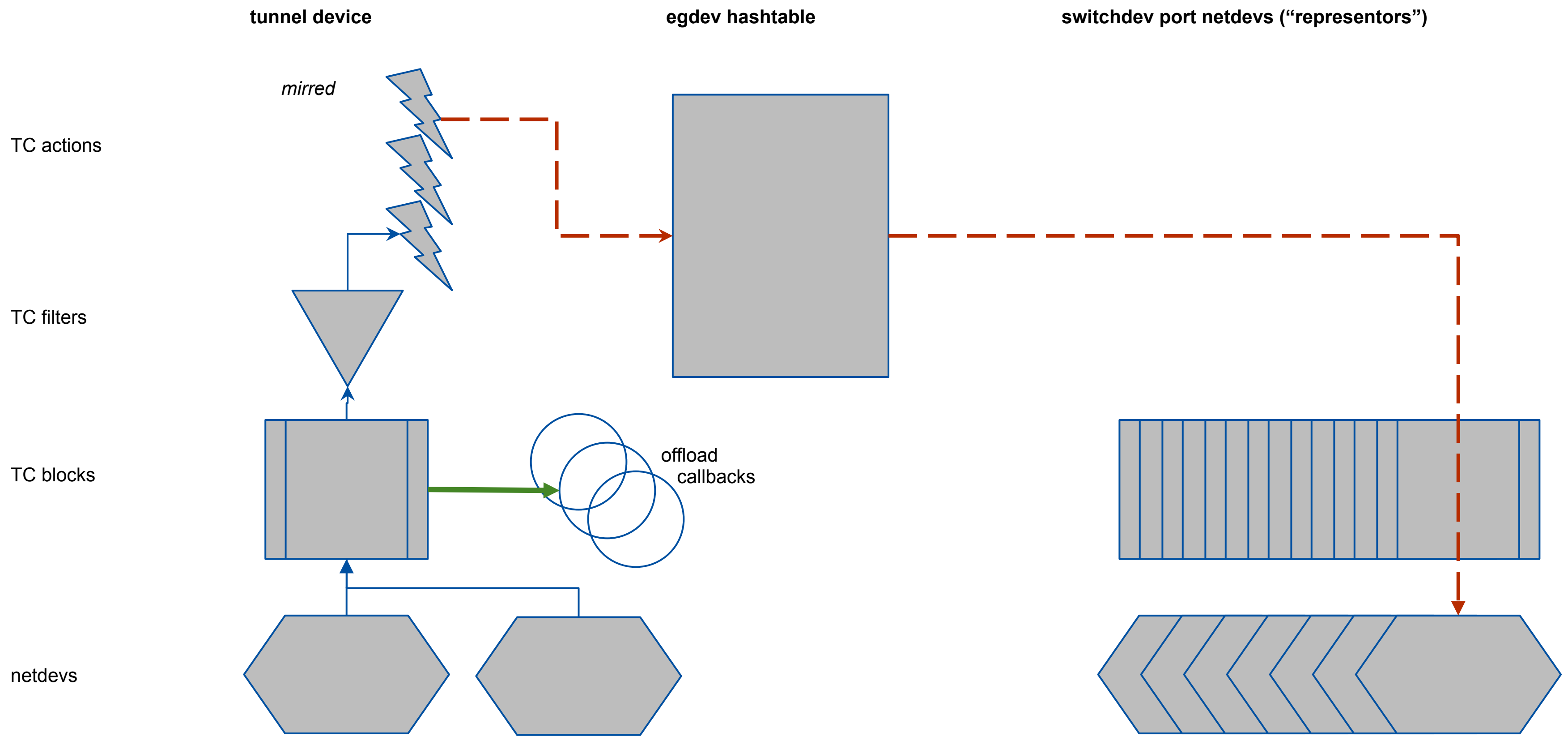




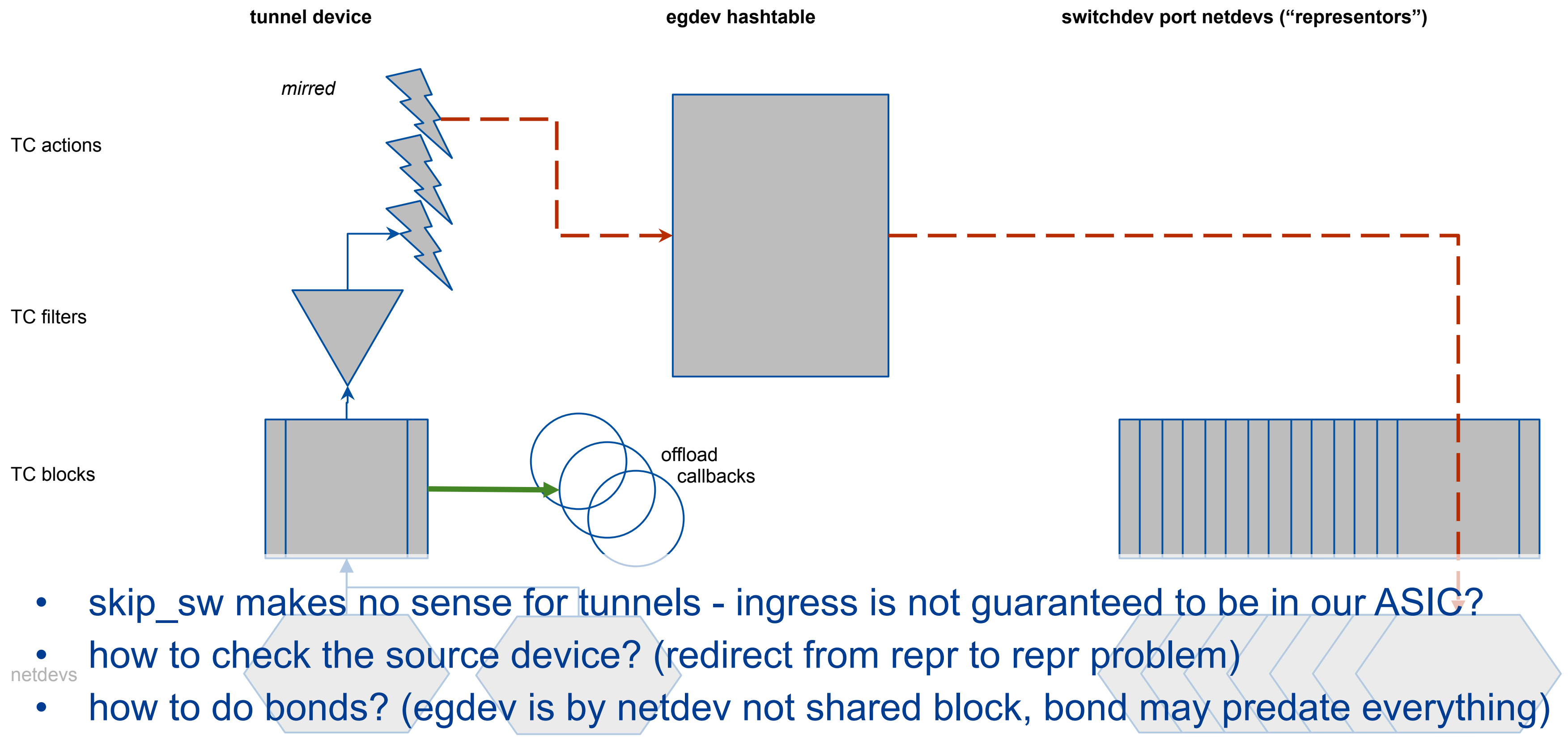
ip link set dev **port0** vf X ...



skip_sw, egdev and tunnels

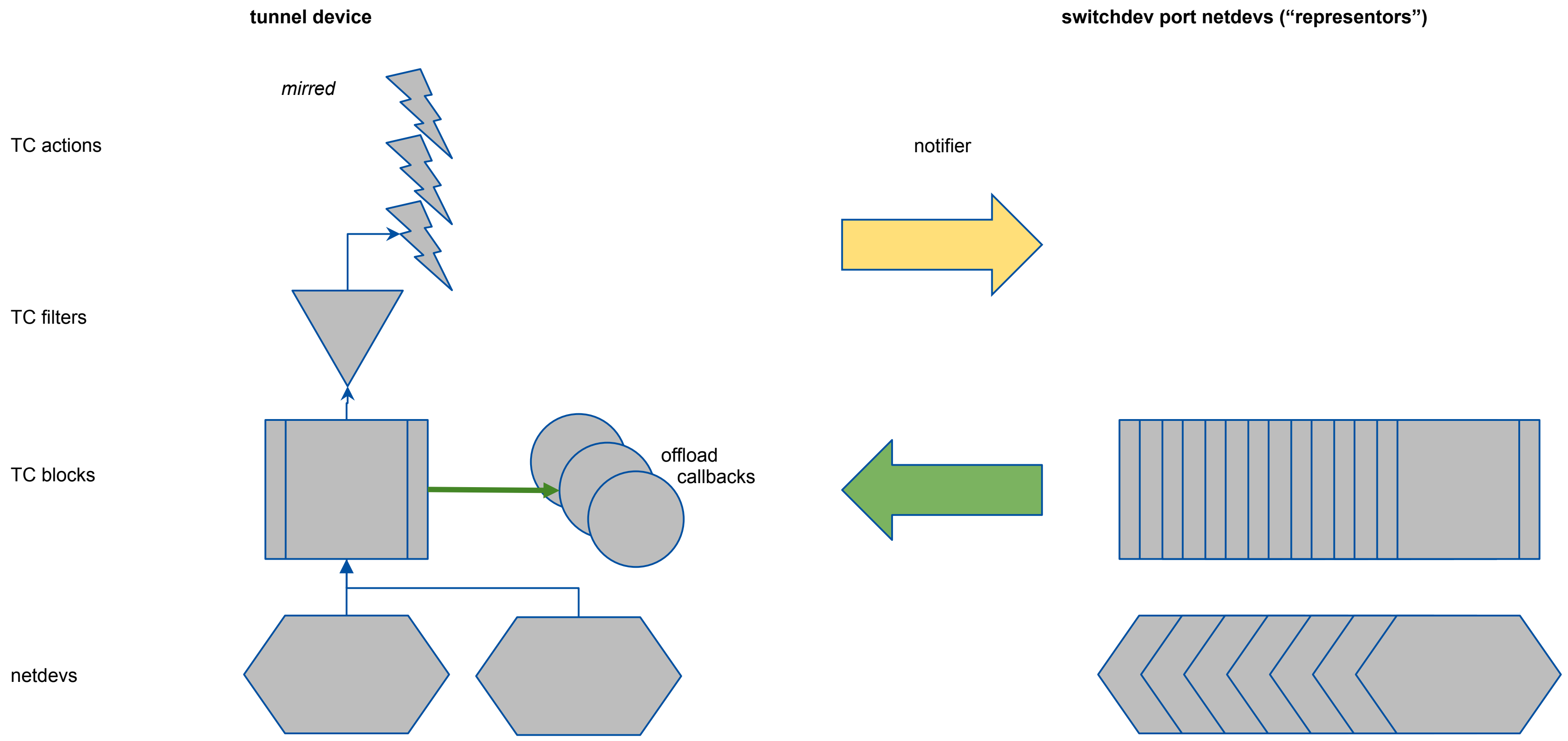


skip_sw, egdev and tunnels



- skip_sw makes no sense for tunnels - ingress is not guaranteed to be in our ASIC?
- how to check the source device? (redirect from repr to repr problem)
- how to do bonds? (egdev is by netdev not shared block, bond may predate everything)
- how to reliably remove the rules (port unsplit; bond destruction)

skip_sw, egdev and tunnels (proposal)



skip_sw, egdev and tunnels (proposal)

- notifier to carry all block bind/unbind events
- drivers can register to blocks of tunnel devices they fancy
- drivers can validate the tunnel device which originates the rule correctly
- no egdev duplicate calls
- drivers can get to redirects to offloaded LAGs trivially
- we can wave goodbye to all the egdev code in the core (yippie!)

Notifier:

- called by the core?
- called by the drivers in .ndo_setup_tc()? (partially solves skip_sw)
- called on the old block?

NETRONOME

Thank you!

Statistics - currently

- .ndo_get_stats64 (aggregate, SW/device mix)
- IFLA_VF_STATS (mlx, cavium)
  - duplicate representors stats
- IFLA_STATS_LINK_XSTATS (bridge)
- IFLA_OFFLOAD_XSTATS
  - IFLA_OFFLOAD_XSTATS_CPU_HIT (switchdev)
- ethtool stats:
  - de facto ban on adding .ndo_get_stats64 duplicates
  - stat groupings appear (veb, port)
  - everyone invents their own names (port, phy, mac)
  - packet counters and event counters

Statistics

- clear indication of dev vs SW stats
- only maintained stats reported
- finer granularity of requests (FW refresh request tunable?)
- nested stats - per queue/prio (XDP stats?)
- control channel statistics (type breakdown or coarse?)
- event/non-port statistics in devlink

What:

- IEEE 803.2.1 stats and other standard stats
- common drivers stats:
  - allocation failed
  - RX page reuse
  - csum, inner csum
  - tso/lro