TC – “chain templates”

- User – limits a chain to certain classifier type and options
  - For example: “cls_flower, dst_mac mask”
- Any filter insertion is checked if it fits the template
- Template provides info to drivers about which match-keys could be inserted in the future
  - Driver can utilize the HW tables in more optimal way

TC - “chain size”

- User - limits the number of filters within a chain
- Driver knows the limit and can prepare the HW tables with the exact size

TC – increase filter insertion rate

- Work of Vlad Buslov
- No parallelism now. RTNL does the serialization – Remove the RTNL dependency
- Use “RTNL_FLAG_DOIT_UNLOCKED”
- Introduce refcounters and small locks to protect individual structs.
- Current work covers all actions and `cls_flower`
- First patchset was submitted upstream, 3 more to go
- The next thing - batching

Devlink - summary

- To be used for configuration whenever netdev is not appropriate to be used as a “handle”
- Can be an ASIC instance
  - pci/0000:00:05.0
  - mdio_bus/fixed-0:1f
- Can be a port
  - pci/0000:00:05.0/1
  - pci/0000:00:05.0/2

Devlink - params

- Work of Moshe Shemesh
- Configuration parameters setting – module params replacement
- Each device registers supported parameters table
- Each parameter can be either generic or driver specific
- User can retrieve data on these parameters by "`devlink param show`"
- User can set new value to a parameter by "`devlink param set`"
- The parameters can be set in different configuration modes:
  - Runtime mode - set while driver is running, no reset required
  - Driver-init mode - set while driver initializes, requires restart driver by “`devlink reload`”
  - Permanent mode - written to non-volatile memory, hard reset required

Devlink – params (examples)

- Generic
  - internal_err_reset - Enable reset device on internal errors. This parameter can be configured on mlx4 either on runtime or during driver initialization
  - max_macs - max number of MACs per ETH port, for mlx4 this parameter value range is between 1 and 128. This parameter can be configured on mlx4 only during driver initialization.
- Driver-specific
  - enable_64b_cqe_eqe - Enable 64 byte CQEs/EQEs when the FW supports this. This parameter can be configured on mlx4 only during driver initialization.
  - enable_4k_uar - Enable using 4K UAR. This parameter can be configured on mlx4 only during driver initialization.

Devlink – regions

- Work of Alex Vesker
- Expose a memory region, typically FW memory
- Allow driver to take a snapshot of a region under certain circumstances
  - User trigger
  - FW catas. event

```
$ devlink region help
$ devlink region show [ DEV/REGION ]
$ devlink region snapshot show [ DEV/REGION ]
$ devlink region snapshot delete DEV/REGION snapshot SNAPSHOT_ID
$ devlink region dump DEV/REGION [ snapshot SNAPSHOT_ID ]
$ devlink region read DEV/REGION [ snapshot SNAPSHOT_ID ]
        address ADDRESS length length
```

Devlink – regions (examples)

- Show all of the exposed regions with region sizes:

```
$ devlink region show
pci/0000:00:05.0/cr-space: size 1048576
pci/0000:00:05.0/fw-health: size 64
```

- Show current available snapshots per region:

```
$ devlink region snapshot show
pci/0000:00:05.0/cr-space: ids: 1 2
pci/0000:00:05.0/fw-health: ids: 1 2
```

- Delete a snapshot using:

```
$ devlink region snapshot delete pci/0000:00:05.0/cr-space snapshot 1
```

Devlink – regions (examples 2)

- Dump a snapshot:

```
$ devlink region dump pci/0000:00:05.0/fw-health snapshot 1
0000000000000000 0014 95dc 0014 9514 0035 1670 0034 db30
0000000000000010 0000 0000 ffff ff04 0029 8c00 0028 8cc8
0000000000000020 0016 0bb8 0016 1720 0000 0000 c00f 3ffc
0000000000000030 bada cce5 bada cce5 bada cce5 bada cce5
```

- Read a specific part of a snapshot:

```
$ devlink region read pci/0000:00:05.0/fw-health snapshot 1 address length 16
0000000000000000 0014 95dc 0014 9514 0035 1670 0034 db30
```

Debug discards

- Spectrum2 HW can provide dropped packet with a reason why it was dropped
- Could be perhaps pushed to userspace via a perf tracepoint?
  - Similar to hwmsg tracing (trace_devlink_hwmsg())

“ethlink” - ethtool alternative

- New port/ASIC features should go to devlink
- New ethernet-netdev specific features should go to a new tool
- “ethlink” - Similar to devlink, using netdev/ifindex as a handle
  - Generic netlink
  - Events by multicast
  - The existing ethtool features are implemented, without need to stick with the existing UAPI structs
- Userspace util would be a part of iproute2 package
- Migration from ethtool
  - Compat layer for existing ethtool UAPI using new kernel interface
    - ethernet-netdev features would be redirected to “ethlink”
    - port/ASIC features would be redirected to devlink
  - Eventually, after everything is implemented by the new tool, ethtool_ops would get removed