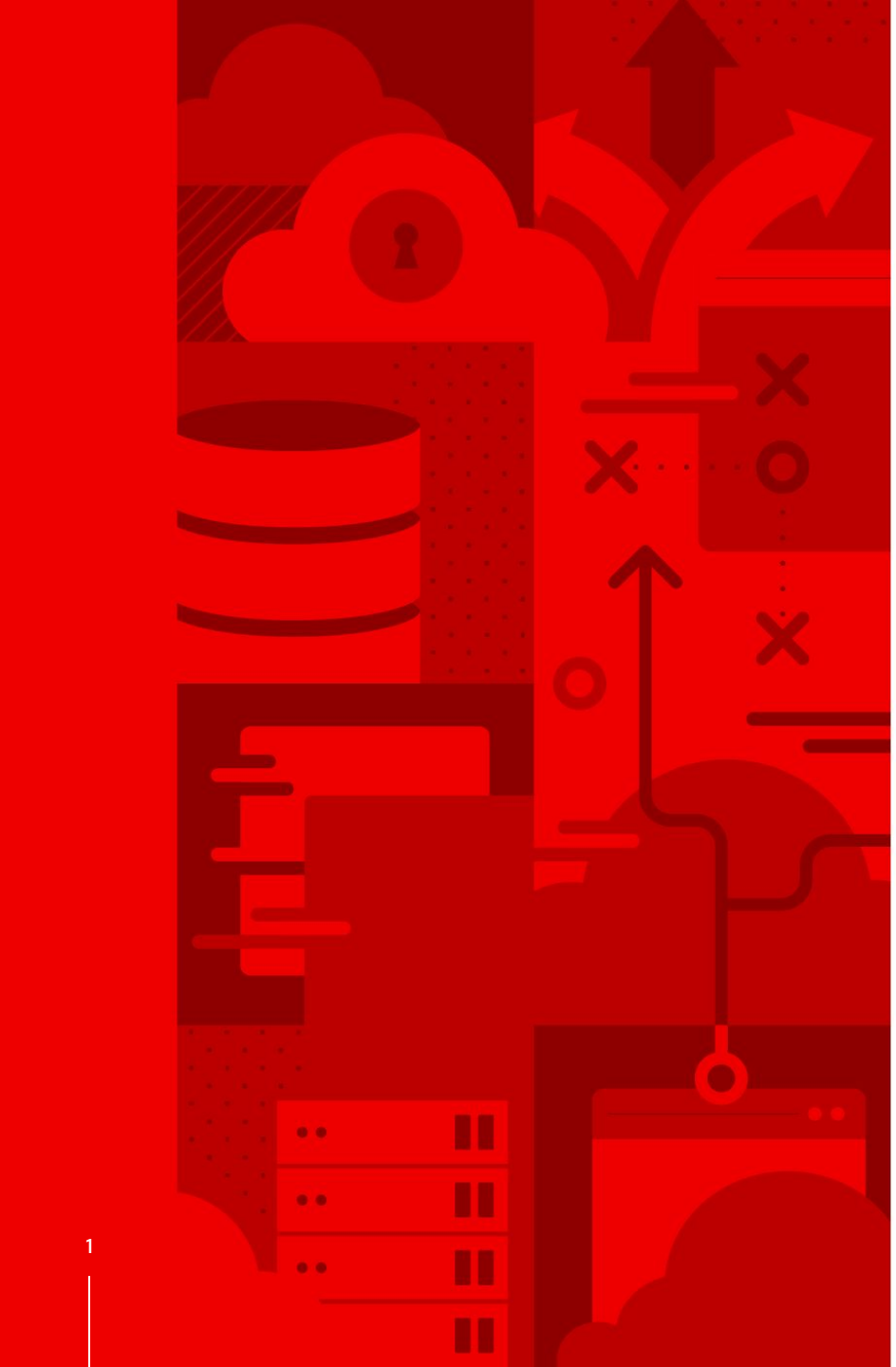


Extending Kernel Hardware Offload (EKHO)

Netlink with YNL

August 2023

Donald Hunter

Billy McFall

Maryam Tahhan

Red Hat

What is Netlink

- A socket protocol for configuring Linux kernel networking
  - User space to kernel, typically NEW, DELETE, SET, GET / DUMP
  - Kernel to user space notifications, typically NEW, DELETE, CHANGE
  - Intra kernel for change notification between subsystems
- Several message families for controlling different networking subsystems
  - NETLINK_ROUTE, including ADDR, LINK, ROUTE, TFILTER, etc.
  - NETFILTER, including TABLE, CHAIN, RULE, etc.
  - OVS including DATAPATH, VPORT, FLOW, PACKET
  - ethtool, netdev, devlink, etc.

What is YNL

A project in the Linux kernel repository to:

- Write specifications for the Netlink APIs
- Write tools (in python) for working with the specifications
- *Generate kernel code* from the spec files for new Netlink features
- Generate user space code from the spec files for new and existing Netlink features

YAML Netlink Slides (Jakub Kicinski LPC 2022, Dublin)

```
tools/net/ynl
├── cli.py
├── ethtool.py
├── lib
│   ├── nlspec.py
│   ├── ynl.c
│   ├── ynl.h
│   └── ynl.py
├── samples
│   ├── devlink.c
│   ├── ethtool.c
│   ├── Makefile
│   └── netdev.c
├── ynl-gen-c.py
└── ynl-regen.sh
```

```
└── generated
    ├── devlink-user.c
    ├── devlink-user.h
    ├── ethtool-user.c
    ├── ethtool-user.h
    ├── fou-user.c
    ├── fou-user.h
    ├── handshake-user.c
    ├── handshake-user.h
    ├── Makefile
    ├── netdev-user.c
    └── netdev-user.h
```

A Netlink Specification

```yaml
# SPDX-License-Identifier: ((GPL-2.0 WITH Linux-syscall-note) OR BSD-3-Clause)
name: netdev
doc:
  netdev configuration over generic netlink.
definitions:
  - type: flags
    name: xdp-act
    render-max: true
    entries:
      - name: basic
        doc:
          XDP features set supported by all drivers
          (XDP_ABORTED, XDP_DROP, XDP_PASS, XDP_TX)
      - name: redirect
        doc:
          The netdev supports XDP_REDIRECT
      - name: ndo-xmit
        doc:
          This feature informs if netdev implements
      - name: xsk-zerocopy
        doc:
          This feature informs if netdev supports AF
      - name: hw-offload
        doc:
         This feature informs if netdev supports XDF
      - name: rx-sg
        doc:
          This feature informs if netdev implements
          support in the driver napi callback.
      - name: ndo-xmit-sg
        doc:
          This feature informs if netdev implements non-linear XDP buffer
          support in ndo_xdp_xmit callback.
attribute-sets:
  - name: dev
    attributes:
      - name: ifindex
        doc:
          netdev ifindex
        type: u32
        checks:
          min: 1
      - name: pad
        type: pad
      - name: xdp-features
        doc:
          Bitmask of enabled xdp-featu
        type: u64
        enum: xdp-act
        enum-as-flags: true
      - name: xdp-zc-max-segs
        doc:
          max fragment count supported
        type: u32
        checks:
          min: 1
operations:
  list:
    - name: dev-get
      doc: Get / dump information about a netdev.
      attribute-set: dev
      do:
        request:
          attributes:
            - ifindex
        reply: &dev-all
          attributes:
            - ifindex
            - xdp-features
            - xdp-zc-max-segs
      dump:
        reply: *dev-all
    - name: dev-add-ntf
      doc:
        Notification about device appearing.
      notify: dev-get
      mcgrp: mgmt
    - name: dev-del-ntf
      doc:
        Notification about device disappearing.
      notify: dev-get
      mcgrp: mgmt
    - name: dev-change-ntf
      doc:
        Notification about device configuration being changed.
      notify: dev-get
      mcgrp: mgmt
```

Red Hat

Netlink Specifications

- Specs get upstreamed to the Linux kernel tree
- New features, for e.g. netdev, are “spec first”
  - Spec is written in YAML
  - ynl-gen-c.py is used to generate kernel code from the YAML spec
- Specs for existing features can be contributed
- Some work-in-progress is in https://github.com/donaldh/linux (tc_flower, netfilter)

```
Documentation/netlink
├── genetlink-c.yaml
├── genetlink-legacy.yaml
├── genetlink.yaml
├── netlink-raw.yaml
└── specs
    ├── devlink.yaml
    ├── ethtool.yaml
    ├── fou.yaml
    ├── handshake.yaml
    ├── netdev.yaml
    ├── ovs_datapath.yaml
    ├── ovs_flow.yaml
    ├── ovs_vport.yaml
    ├── rt_addr.yaml
    ├── rt_link.yaml
    └── rt_route.yaml
```

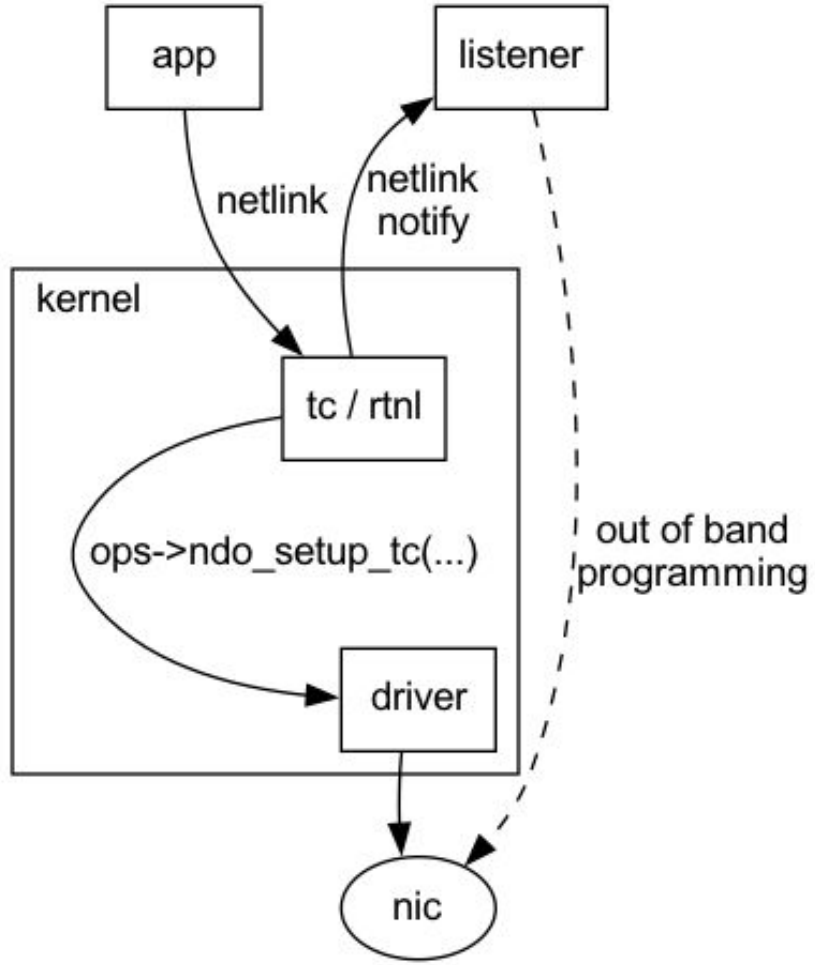
Using YNL

```
./tools/net/ynl/cli.py \
     --spec Documentation/netlink/specs/ovs_datapath.yaml \
     --dump get --json '{ "dp-ifindex" : 0 }'

[{'dp-ifindex': 3,
  'masks-cache-size': 256,
  'megaflow-stats': {'cache-hits': 1658813,
                     'mask-hit': 50647298,
                     'masks': 2,
                     'pad1': 0,
                     'padding': 0},
  'name': 'ovs-system',
  'stats': {'flows': 6, 'hit': 6529458, 'lost': 12, 'missed': 2118584},
  'user-features': {'dispatch-upcall-per-cpu',
                    'tc-recirc-sharing',
                    'unaligned'}}]
```

Recap: Mirroring kernel networking state

- All configuration of the Linux networking stack from user space is done using netlink.
- It is possible to listen for all application-driven configuration changes by registering for netlink notifications across all the relevant netlink message families.
- A combination of get bulk (dump) and notifications makes it possible to stay in sync.

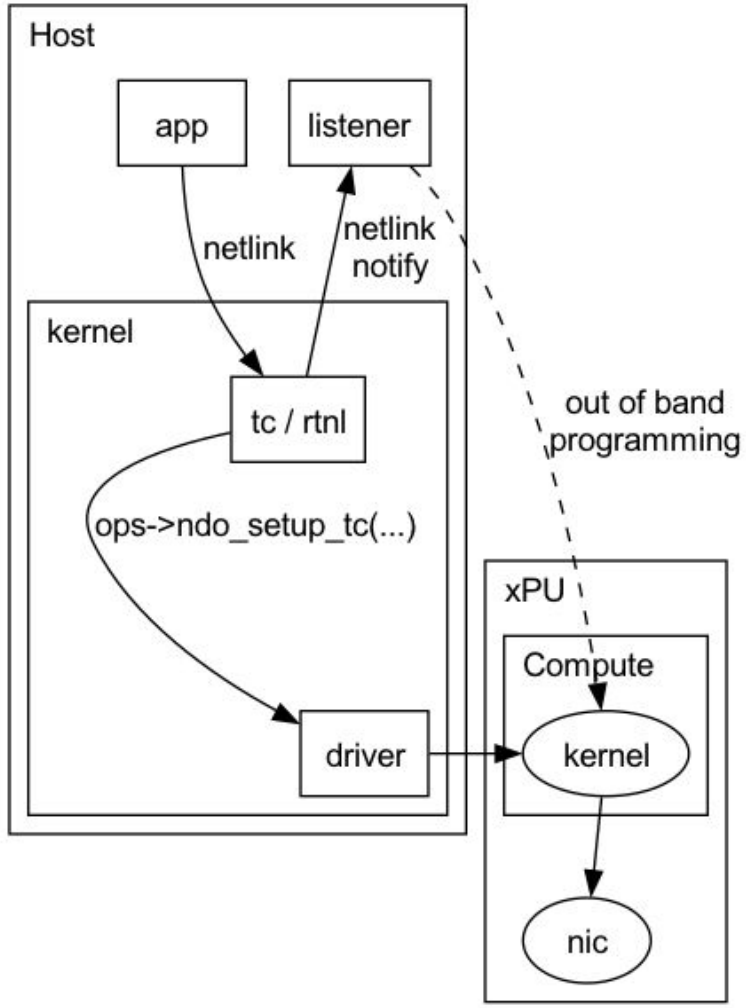


Red Hat

PoC – Mirroring networking state to an xPU

Selectively offload networking to an xPU:

- Receive Netlink notifications from the kernel on the host
- Filter and transform into the desired offload configuration for the xPU
- Push configuration updates to the kernel on the xPU
- Validate feasibility of mirroring state from kernel to hardware pipeline on an xPU



Red Hat

EKHO – A Red Hat PoC

Exploring “Extending Kernel Hardware Offload”. One goal of this PoC is to:

- Use YNL to mirror networking state from kernel to kernel to simulate kernel to hardware
- For the purposes of this PoC, use Netlink specs *and* protocol
- Validate:
  - Selectively offloading networking configuration to an xPU
  - Maintaining compatibility with existing user space applications
- Identify:
  - Bugs and feature gaps in Netlink
  - Semantic challenges

YNL Netlink Specs and OPI-API

- The Netlink specs provide a machine readable definition of each Netlink protocol family
- Potential for code generation or transformation into other schema formats
  - E.g. re-use as base definitions for protobufs
- Potential for reusing the Netlink message specifications with other transport protocols
  - E.g. Netlink messages over gRPC for “Linux Networking” in OPI

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat