
facebook

AF_XDP user experience

Jonathan Lemon

bsd@fb.com

Review of AF_XDP

Exactly what the hell is this thing?

- Focused on low latency, low overhead handling of packets.
- Can be viewed as linux alternative to DPDK.
- Drivers in kernel, filters in bpf, user allocated memory.
- Successor to AF_TPACKET v2, v3
- Limitations, AKA "benefits":
  - intentional bypass of most kernel features

Use cases

Solution in search of a problem

- QUIC, custom steering rules.
- High speed presence/counter updates.
- Prototype development.
- Load testing.

Details

... the devil is in there somewhere.

- AF_XDP application would typically be main application, but not the only thing running on the box, so it would need to coexist with normal traffic.

- Would possibly be nice to allow multiple AF_XDP applications; this would require multiple bpf filters for a single netdev.

Issues

.. so is the fly in my soup.

- Ease of use, understanding, deployment & configuration
- Queue configuration, bundling and specifications.
- Co-existence with normal traffic without penalties.
- Uniform behavior across drivers.

RSS/classification separation

Redirect traffic to a specific RSS context:

```
ethtool -X eth0 context new 2
ethtool -N eth0 flow-type udp6 dst-port 4242 context 2
```

Redirect traffic to a specific queue:

```
ethtool -N eth0 flow-type udp6 dst-port 4242 action 30
```

Not working together.

Idealized experience

- Receive object is created
  - Either by driver or by user, matching parameters
- Object is attached to context (RSS distributor)
  - Contexts can be default, or created by user
- bpf redirects to context
  - Substitute simple hardware filter in place of bpf to start.

packets -> bpf -> distributor -> queues

Reasonable API available to user space.

Goals

- Discussion on object management (RX, TX, NAPI)
  - Creation, properties, referencing
- How these are bound to resources (device, cpu, irq, numa)
  - Requests, query capabilities

- Automated lifetime (or part of object management)
- Reasonable unified API (for both user & kernel code)

Small Improvements

... little by little

- BTF type for xsk, simplifying xskmap lookup, eliminating need for shadow map.
- reuseq stack in front of fill queue, for kernel pushback.
- zero-copy transmit for XDP_TX decision on RX path.

Future Vision

- Moar offloading (BPF in hardware?)
- Would AF_XDP sub-device be a better solution to these problems?
- Ideally would be great to have NIC vendors to focus on NIC features, not drivers.


facebook