LSF/MM/BPF

Daniel Borkmann, Isovalent

May 2022

Topic 1: uveth & tc BPF rework

(Skipping tc BPF rework in here since discussed on Mon already)

Goal: Pod networking with same efficiency as host

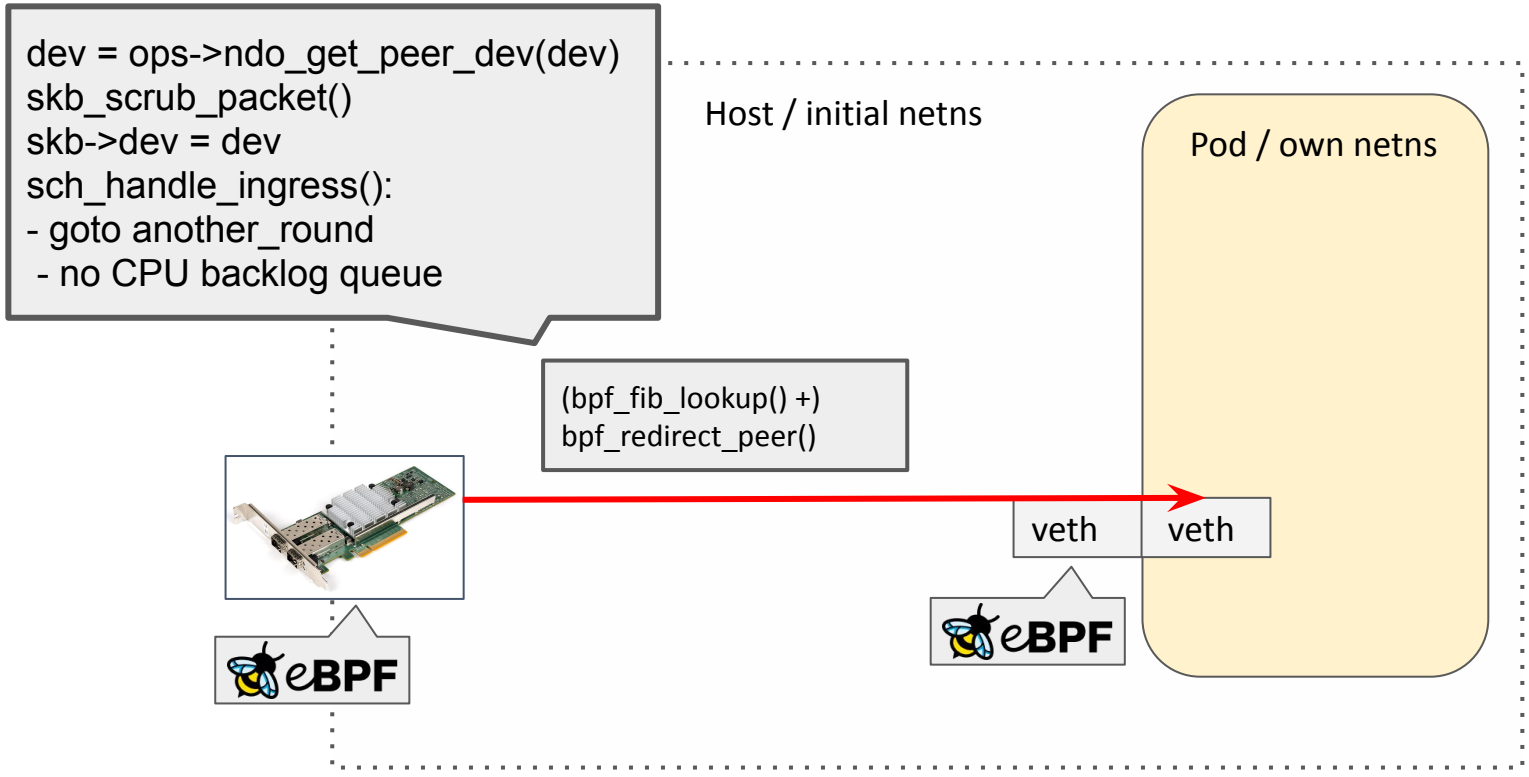
Steps towards this direction so far:

- Retaining of skb->sk across netns switch
- BPF host routing to only rely on tc layer for forwarding instead of upper stack
  - bpf_redirect_peer() / bpf_redirect_neigh()
  - bpf_fib_lookup()
- skb->tstamp preservation

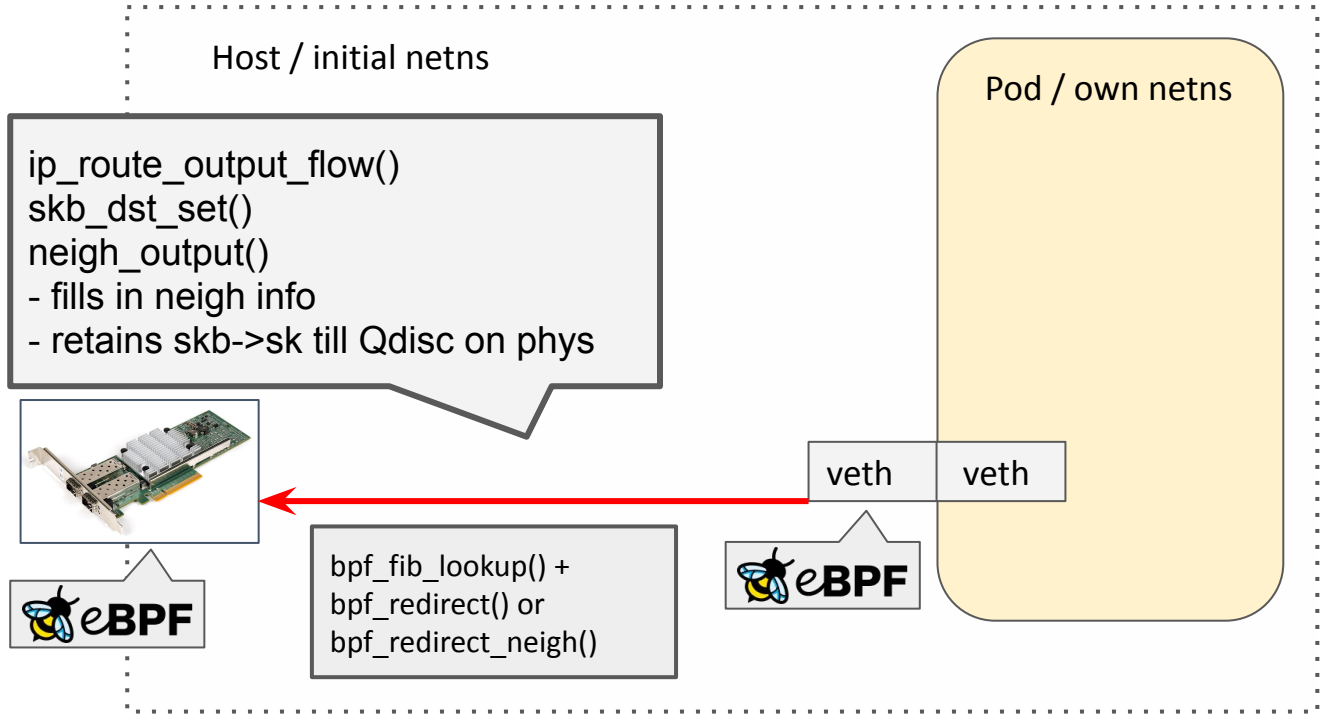
Common theme:

- Holding skb->sk all the way until phys driver's TX completion (e.g. TCP TSQ feedback)
- Retaining important skb meta data instead of scrubbing
- Efficient namespace switch without detour through backlog queue

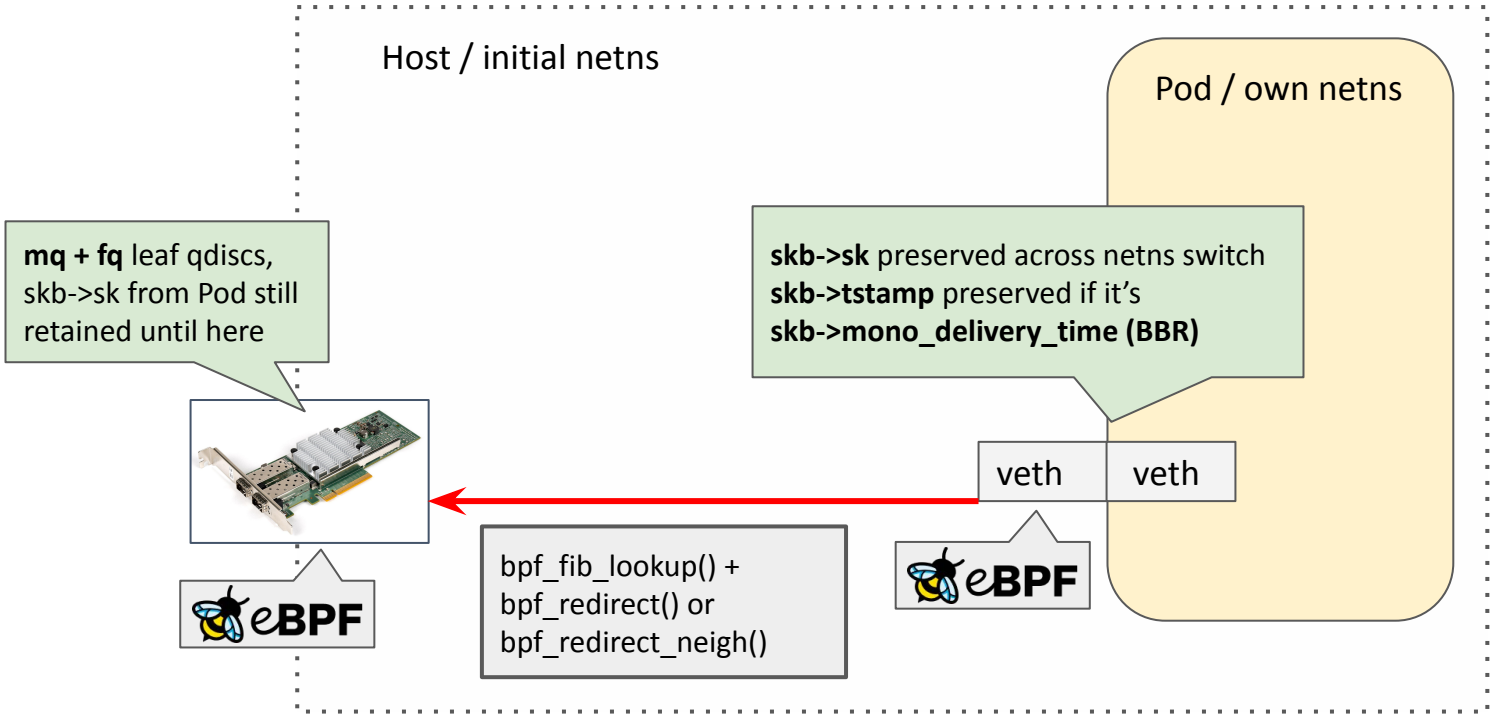
BPF datapath, quick recap



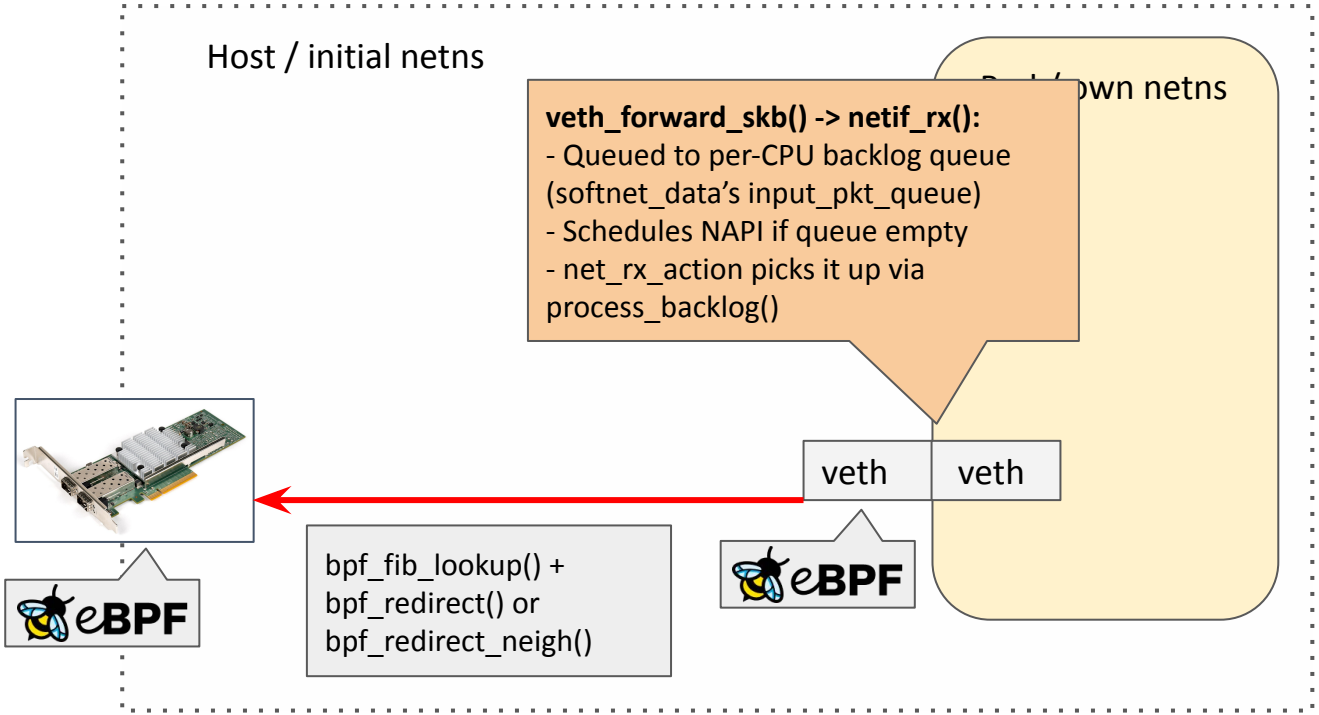
BPF datapath, quick recap



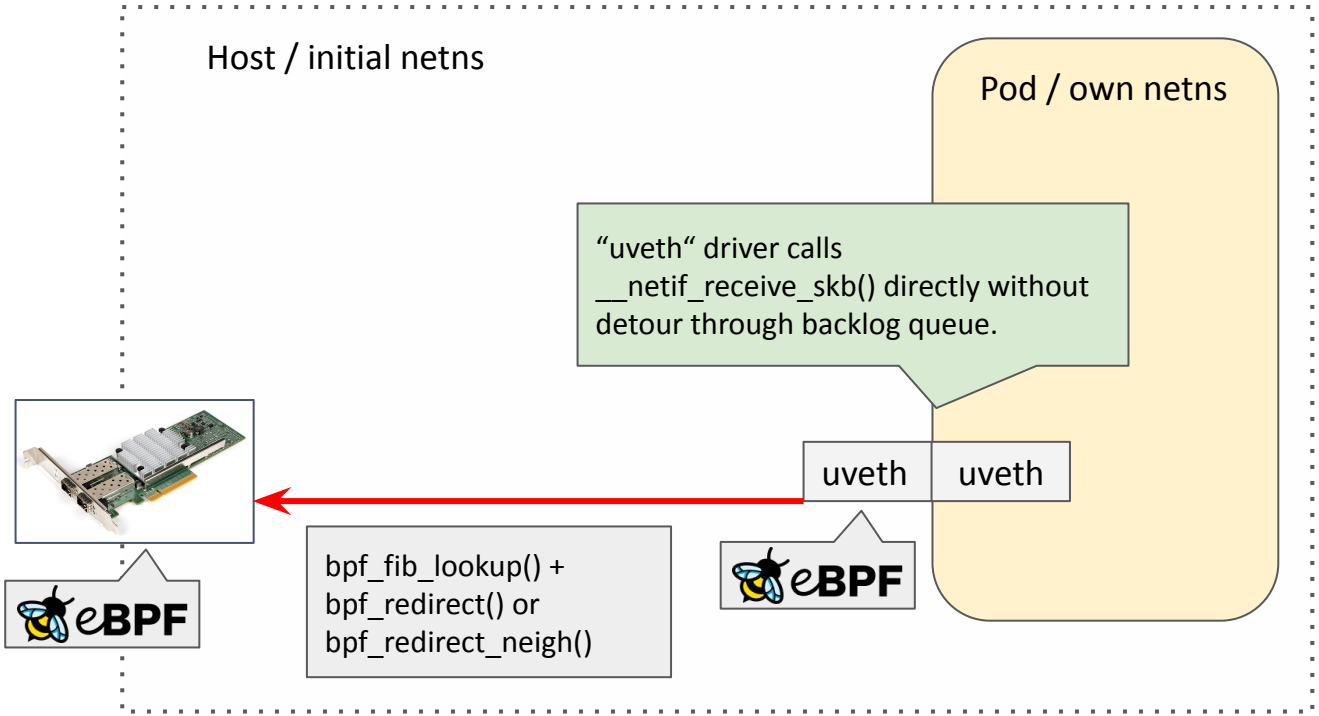
BPF datapath, quick recap



BPF datapath, next step

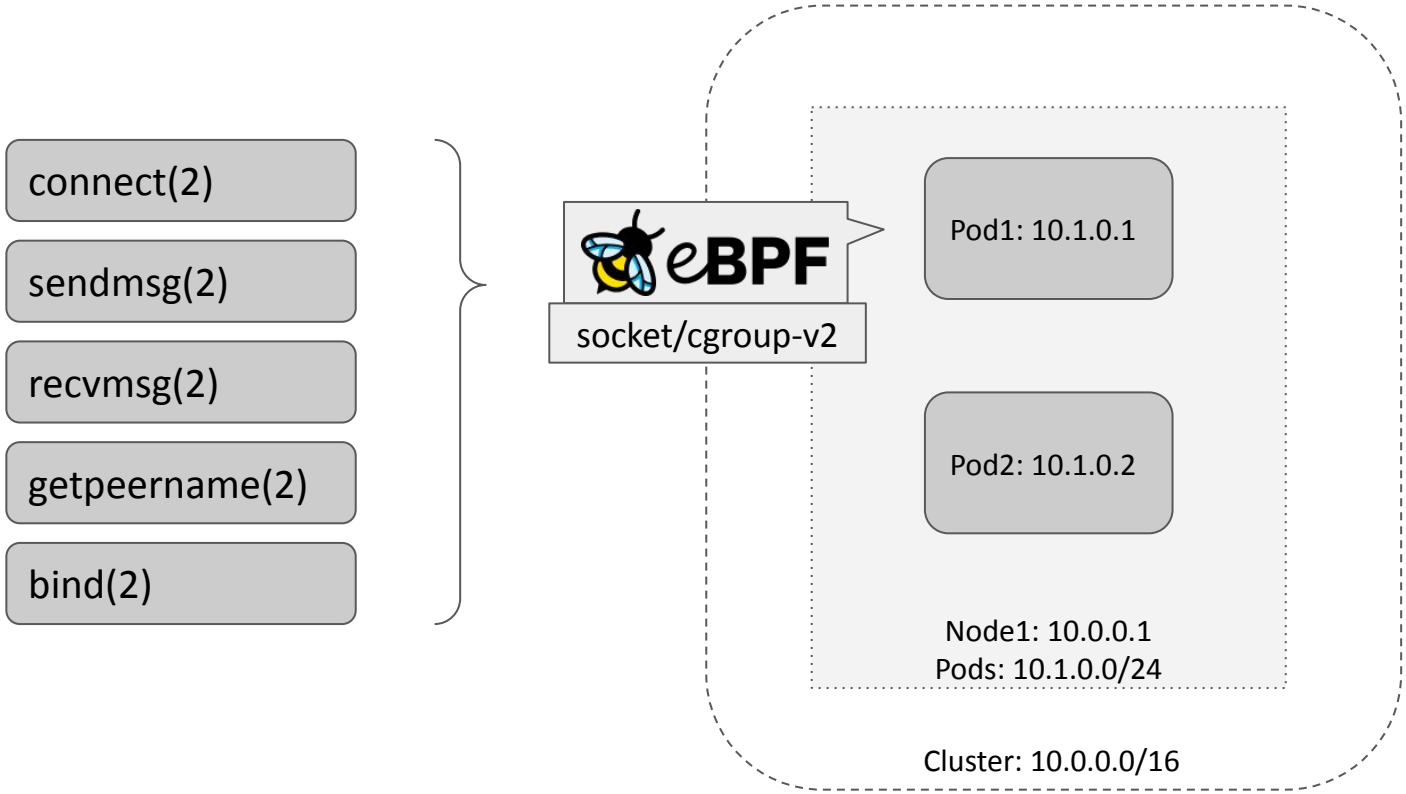


BPF datapath, next step

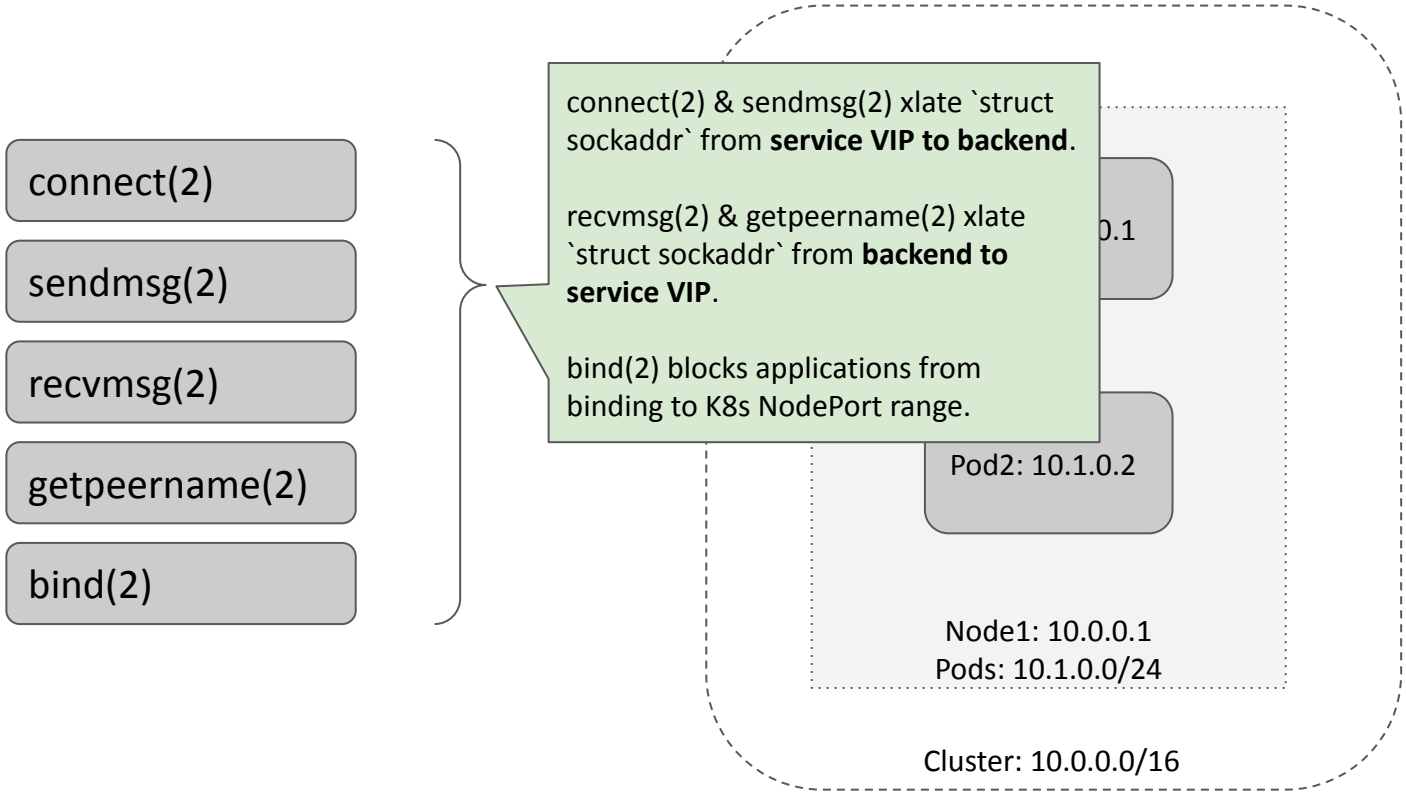


Topic 2: Socket hooks for connected UDP/TCP

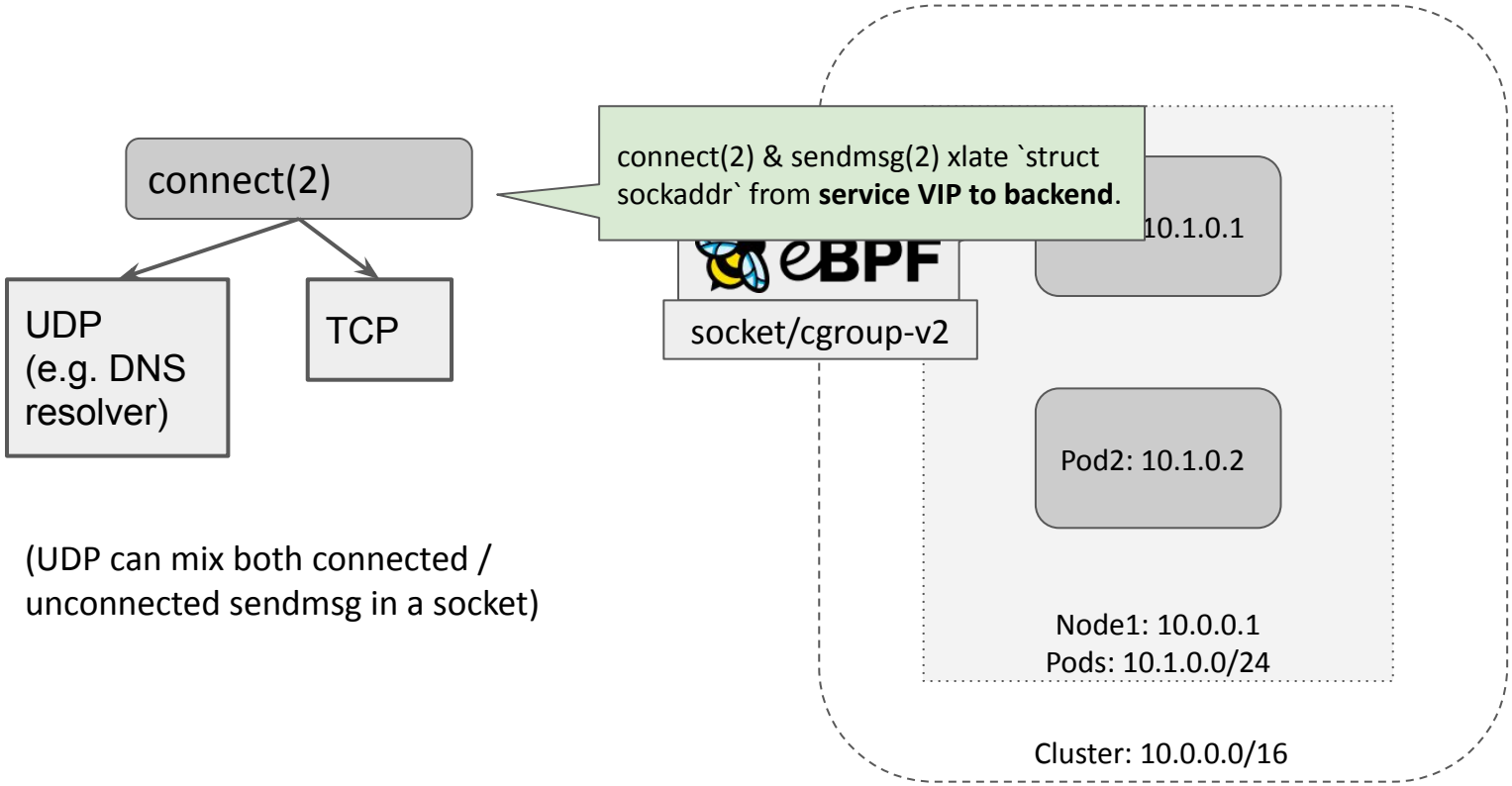
BPF E/W load-balancer, quick recap



BPF E/W load-balancer, quick recap

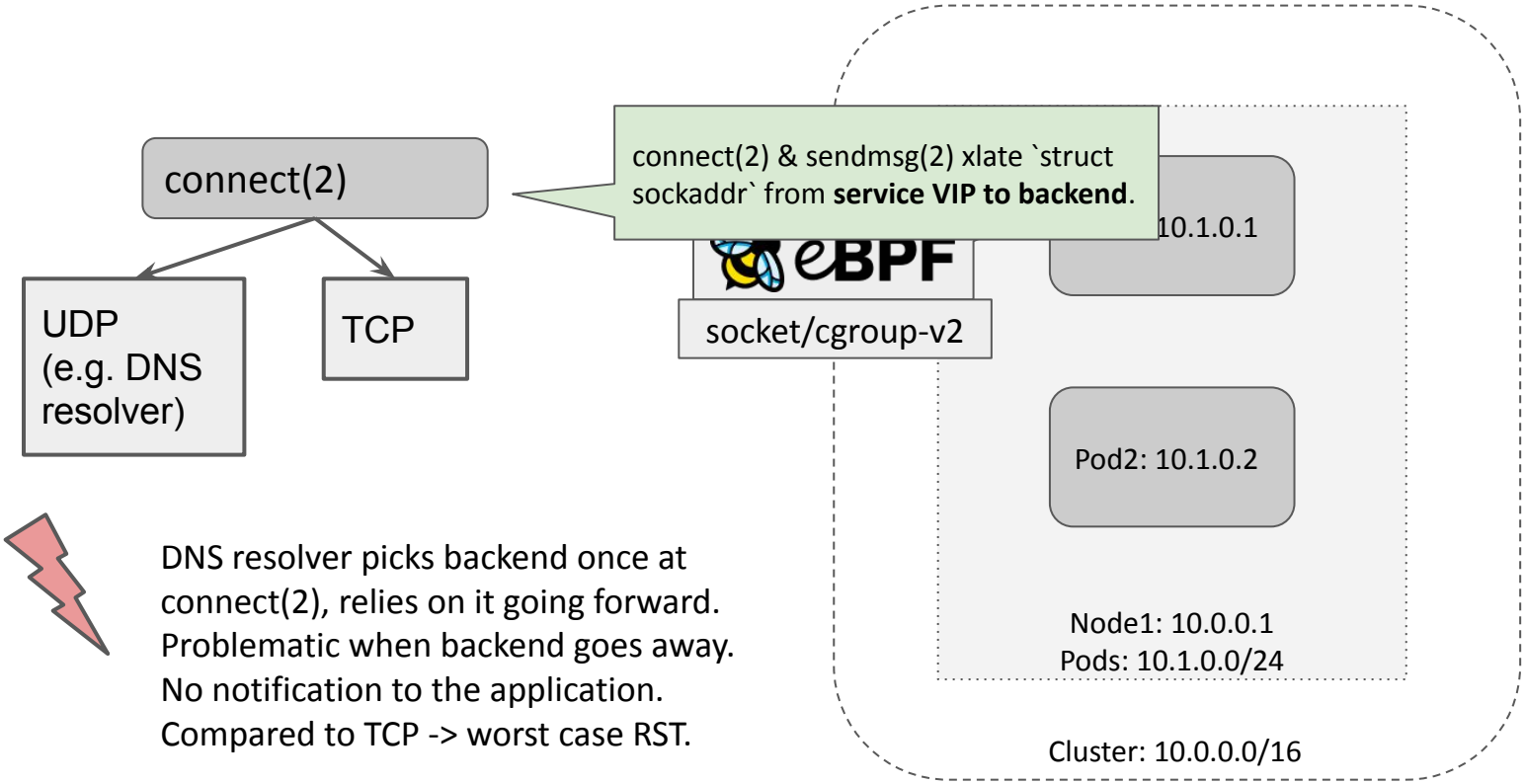


BPF E/W load-balancer, next steps



(UDP can mix both connected / unconnected sendmsg in a socket)

BPF E/W load-balancer, next steps



DNS resolver picks backend once at connect(2), relies on it going forward.
Problematic when backend goes away.
No notification to the application.
Compared to TCP -> worst case RST.

BPF E/W load-balancer, next steps

```
if (cgroup_bpf_enabled(CGROUP_UDP4_SENDMSG) && !connected) {
        err = BPF_CGROUP_RUN_PROG_UDP4_SENDMSG_LOCK(sk,
                                    (struct sockaddr *)usin, &ipc.addr);
        if (err)
                goto out_free;
        if (usin) {
                if (usin->sin_port == 0) {
                        /* BPF program set invalid port. Reject it. */
                        err = -EINVAL;
                        goto out_free;
                }
                daddr = usin->sin_addr.s_addr;
                dport = usin->sin_port;
        }
}
```

BPF E/W load-balancer, next steps

Proposal: new hook for **sendmsg** for case when socket is connected

- connected UDP
- connected TCP

Input context is similar to bind(2) hook: **struct bpf_sock *ctx**

Allows for:

- Using socket cookie or socket local storage to gather service/backend ID
- Can check if backend is still alive in backend BPF map
- If not:
  - UDP: Calls a new **bpf_connect()** hook to enforce new backend selection & caching of new dst
  - TCP: Return **ECONNRESET** to application, or new helper for destroying socket (similar to ss tool's SOCK_DESTROY)