

BPF updates

recent developments in BPF

- s390 jit is almost ready
- bpf backend in llvm has been upstreamed
- bpf+tracing is being discussed
 - bpf+syscall and bpf+kprobe is looking good
 - bpf+tracepoint needs more time (eventually can be used for tcp stats)

recent developments in BPF

- two teams in huawei are trying to use bpf+tracing
- sysdig made a tool inspired by tracex3 example
- Brendan@netflix is playing with bpf+tracing
- bpf+ovs, bpf+tc are being discussed
- bpf+seccomp is waiting for 'unprivileged' mode

BPF concerns

- bpf for modules
- three different attachment points in ovs+bpf
 - bpf as an action on flow-hit
 - bpf as packet parser before flow lookup
 - bpf as fallback on flow-miss
- two attachment points in tc+bpf
 - cls – packet parser and classifier
 - act – action ? packet rewrite ?
- C sometimes is too generic for programs

common building block for BPF+networking ?

- need to have a common bpf+networking building block
- what do we want it to do?
 - parse packets
 - modify packets
 - select next networking block
- use restricted C or 'network oriented language' for programs

building block for BPF+networking ?

- can bpf be a net_device ?
 - netlink/ndo_do_ioctl to install bpf_prog_fd
 - ndo_start_xmit will execute bpf program with given skb
 - registering rx_handler will pass processed skb back to ovs/tc?
 - metadata can be passed in cb
 - ovs action and flow miss can work with start_xmit/rx_handler model
- ingress ovs needs such bpf_dev to have a port:
 - add 'tap' to bpf_dev, add bpf_dev to vport