configurable packet steering with nft

Florian Westphal
4096R/AD5FF600 fw@strlen.de
80A9 20C5 B203 E069 F586
AE9F 7091 A8D9 AD5F F600

Red Hat

netdev 2.1, Montreal, April 2017

rfs, rps

- RPS: Receive packet steering: distribute packets based on rxhash
  - select cpu to process packet
  - place skb on a cpu backlog queue
  - signalling via IPI
  - cpu selection via sysfs
  - some heuristics (flow limit)
- RFS: Receive Flow Steering
  - uses RPS mechanisms
  - maps flows to cpu where application is running

nft meta

- used to match in/out interface, skb mark, etc
  can also match cpu
- instead of "`meta cpu 2`" ... allow to set it: `meta cpu set 2`
- return `NF_STOLEN`
- other cpu will continue processing that skb
- could e.g. be used to split away some packets e.g. for ipsec, forwarding, ... in a configureable fashion

still early toy stage

poc details

- per cpu data: work queue, `skb_array`, nft hook state
- multiple producers (rx processing), one consumer
- skb + state gets placed in skb array of desired cpu
- then uses `queue_work_on` for that cpu
- work queue dequeues all skbs
- then reinject them into stack (calls `state->okfn(.. skb ..)`

todo

- get numbers
- depending on positive results:
- resume processing in next netfilter hook (table) instead of okfn
- ... could generalize nfqueue infrastructure for this