Network Function Offload

Networking Workshop @LKS
Aug 19, 2014
<hadi@mojatatu.com>

Context

- On going desire for offload getting more exposure because of virtualization efforts
- 2 day conference video/calls in April
  - Agenda can be seen here:

    https://linux.cumulusnetworks.com/offload-discussion-1/agenda.txt
  - Slides at: https://linux.cumulusnetworks.com/offload-discussion-1/
- Effort to harmonize disparate attempts for offloading
  - NICs offloading functions for virtualization
  - The home router world (openwrt leading)
  - The Large device world

Agreements

- MUST Use Linux APIs
  - No vendor APIs
    - We know how to do netlink
    - We may have to add new functionality
- Current tools MUST continue to work, ex:
  - Nftable/iptables
  - iproute2 cousins (route, link, arp, xfrm, tc, bridge etc)
  - Legacy tools like route, ifconfig
  - Etc
- Things that look like ports will appear as netdevs

Challenges

- Host vs Network centric view of the world
  - To the NIC vendors everything is a server endpoint
  - To the middle box function vendors everything is a transient point
  - We need to support both and not optimize for one
    - Something we already do well?

Challenges

- Network Functions will have mismatches/quarks in capabilities and function
  - Some popular offload network functions:
    - Bridging/switching, QoS, IPSEC, L3 forwarding, stateless ACL
  - Is VMDQ really bridging?
  - A bridge implemented by a Realtek chip does not offer the same functionality as a brcm tridentII
  - An L3 function that integrates FIB+NH selection vs separate tables

Challenges

- Network Functions will have mismatches in capacities
  - Large vs Small table sizes
  - Large vs small wire processing capacities

Challenges

- Network Functions will have different desired use cases (implying varying policies)
  - Do you really want the kernel to make autonomous decisions or do we need user space helpers?
    - Example: What kind of beast is MacVLAN these days?
      - It self-contains lotsa functions that exist elsewhere
    - Example: NIC multicast MAC tables first use up all the hardware entries and then enforce promisc
      - That is hard coded policy IMO (need to be able to define behavior)
  - If we agree that we need user space helpers, should we define user space APIs?
    - Large ASIC vendors happy at the expense of openness

Challenges

- To some folk *everything is a table*
  - Not pragmatic at all
- Offloaded Network Functions will have different implementation-to-model maps
  - Some things are trees
  - Some things are tables
  - Some things are scalars

Challenges

- Some offload network functions are at the middle of packet processing graph
  - i.e singular offloaded function in the middle of path where all other functions run in s/w
    - Example crypto hardware offload in ipsec processing