Checksum offload and UDP encapsulation protocols

https://tools.ietf.org/html/draft-herbert-remotecsumoffload-02

https://tools.ietf.org/html/draft-herbert-vxlan-rco-01

Tom Herbert <tom@herbertland.com>

Checksum offload

- Offload RX or TX L4 checksum to NIC
  - TCP/UDP/ICMP etc.
  - Performance benefits
- Protocol /non-protocol agnostic methods
- Encap allows for >1 checksum per packet
- Goal: **No full packet host csum calculation**

TX checksum methods

- NETIF_HW_CSUM
  - Initialize checksum to pseudo header csum
  - Input to device *start* and *offset*
  - HW checksums from start to end of packet and writes result at offset
- NETIF_IP_CSUM (legacy)
  - HW can only checksum with certain protocol hdrs
  - Typically UDP/IP and TCP/IP
  - HW handle pseudo hdr csum also

RX checksum methods

- CHECKSUM_COMPLETE
  - HW returns csum calculation across whole packet
  - Host uses returned value to validate checksum(s) in the packet
- CHECKSUM_UNNECESSARY (legacy)
  - HW verfies and returns “checksum okay”
  - Protocol specific, HW needs to parse packet
  - csum_level allows HW to checksum within encapsulation, multiple checksums

Leveraging UDP checksum offload

- Probably every deployed NIC supports simple UDP checksum for TX and RX
- Newer NICs support offload of encap csum
- Solution: **Enable UDP checksum for encap**
  - Receive: checksum-unnecessary conversion
  - Transmit: local/remote checksum offload

Checksum tricks

- Checksum unnecessary conversion
  - Device returns “checksum unnecessary” for non-zero outer UDP checksum
  - Complete checksum of packet starting from the UDP header is ~pseudo_hdr_csum
- Local checksum offload
  - Can infer outer checksum value when an inner checksum sums to zero
  - Useful with devices that provide NETIF_HW_CSUM
  - Supports arbitrary number of outer checksums
  - Consequence: **No need for HW to support more than one csum calculation per packet!**

Remote checksum offload (RCO)

- Defer TX checksum offload to remote
- Encapsulation header with *start* and *offset* data referring to inner checksum (*protocol*)
- Offload outer UDP checksum and send
- At receive
  - Do what device does: determine checksum from start to end of packet and write to offset
  - Already have complete checksum so we can easily find this
  - Write checksum into packet, validate like normal

RCO & VXLAN questions

- Supported in Linux
  - One header bit
  - Lower ordert eight bits of VNID for encoded start/offset values
- Is this the intended use of those fields?
- How do we make this official? (currently RX config option)
- Same question for VXLAN-GPE...