GRO: Surviving 10Gbp/s with Cycles to Spare

Herbert Xu
Red Hat Inc.

Background

- Ethernet bandwidth from 10Mb/s to 10Gb/s.
- MTU (Maximum Transmit Unit) 1500 bytes.
- Packet rate from 833pps to 833000pps.
- CPU speed has flatlined.
- Packetisation overhead now a bottleneck.

Raising MTU

- Larger MTUs through Jumbo Frames.
- Standard jumbo frames are only 9000 bytes.
- Lowest MTU over whole path applies.
- PMTU discovery determins MTU.
- PMTU discovery works poorly over Internet.

TCP Segmentation Offload

- Content providers biased towards sending.
- TSO raises MTU within host to 64KB.
- Resegmentation before wire.
- Sufficient for 10GbE.
- IPv6 jumbograms for MTUs above 64KB.
- Linux has Generic Segmentation Offload.

Large Receive Offload

- NIC still receives ~1 million packets.
- LRO merges packets upon entry into stack.
- Larger internal MTU as TSO.
- Widely supported by 10GbE drivers.

LRO Problems

- Packet merging doesn't preserve all state.
- Incompatible with packet forwarding.
- Incompatible with virtualisation.
- LRO limited to TCP over IPv4.

Generic Receive Offload

- GRO restricts what can be merged:
  - Identical MAC header.
  - Only certain IP header fields can differ.
  - Only certain TCP header fields can differ.
- Merged packet can be resegmented losslessly.
- GRO reuses GSO infrastructure.

GRO Status Quo

- Supported by many (but not all) 10GbE drivers.
- Complete conversion of remaining drivers.
- Address any performance regressions.
- Eventual removal of LRO.

Future Work

- Generic flow-based merging:
  - Linked list of skb's per IP flow.
  - Merging UDP and other protocols.
- Reuse hardware receive hash in GRO.
- Emulate multiqueue receive in software.

Questions