Linux Networking State

David S. Miller, Red Hat Inc.

Proceedings of netdev 0.1, Feb 14-17, 2015, Ottawa, On, Canada

TCP

Per-flow Pacing

Fast Open

Dynamic TSO Sizing

Datacenter TCP

TCP Small Queues

Advanced TCP statistics (web10g)

Queueing Minimization

- Socket queues
  - TCP Small Queues
- QDISC queues
  - Fair Queue Packet Scheduler (per-flow pacing)
  - Effect on TCP Timestamps
- Device queues
  - Byte Queue Limits

Checksumming

Partial checksum propagation through encaps

Power of CHECKSUM_COMPLETE

Remote Checksum Offload

Encapsulation meta-data which allows deducing

the checksum of the encapsulated protocol

Switch Offloading

Bridge forwarding

IPV4/IPV6 route forwarding

nftables

Offload Policy Part 1

Example: ip route add xxx

If we are offloading ipv4 forwarding to hardware, and the device indicates that this new route cannot fit in it’s hardware tables, what do we do?

Policy Part 2

Option 1:

Do not install the route and return an error.

Option 2:

Uninstall all hardware routes and do all forwarding in software.

Option 3:

Use hw as much as possible w/sw fallback

Policy Guiding Constraints

It must by default be %100 transparent to the user.

This means no errors when exceeding hw capacity.

By this definition options #2 and #3 are permissible

But… we can provide facilities for people who want to do something sophisticated in this situation.

Multiple Offload API Tracks

Direct bridge FDB and ipv4-route device operations (Scott Feldman and Jiri Pirko)

Flow API (John Fastabend).

And if a third set of interfaces is proposed, that’s OK too.

Eventually with enough experience things will converge.

What Really Matters

A clear plan, with well defined constraints.

Unambiguous reasons for each and every constraint.

Someone will be unhappy with the design we come up with, this is inevitable. So we must be able to explain our design decisions precisely.

rhashtables

Resizable hash tables using RCU locking

Current users: netlink sockets and nftables

Use for TCP sockets in the future

TX Overhead Mitigation

skb->xmit_more

Decreases number of doorbell rings per packet

Especially important for virtualization devices

Enhanced with bulk dequeue support in packet scheduler

Busy Polling

Alternative to blocking at recvmsg() time

If recvmsg() finds socket receive queue empty

Call into device driver and poll for RX packets

If any found, feed into networking stack

recvmsg() pulls any received data into userspace

Memory Allocation Batching

Networking stresses SLAB/SLUB

Unbalanced RX/TX allocation/free patterns

Allocation overhead can exceed the time budget we have for processing small packets at 10GB wire rate

qmempool developed as an experiment to see what allocation batching can do

SLAB/SLUB extended to have batching interfaces

Thanks

Linus Torvalds

Jamal Hadi Salim

And in advance, I'd like to thank the first hardware vendor to merge a hw switching driver upstream. You will be a true trail-blazer.