Packet Pacing design highlights

Meny Yossefi

2.10.2015

FreeBSD

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™

Agenda

- Introduction
- Kernel and User space
  - Main flow
  - Implementation
- Mellanox driver - Highlights
- Feature Characteristics
- Demo

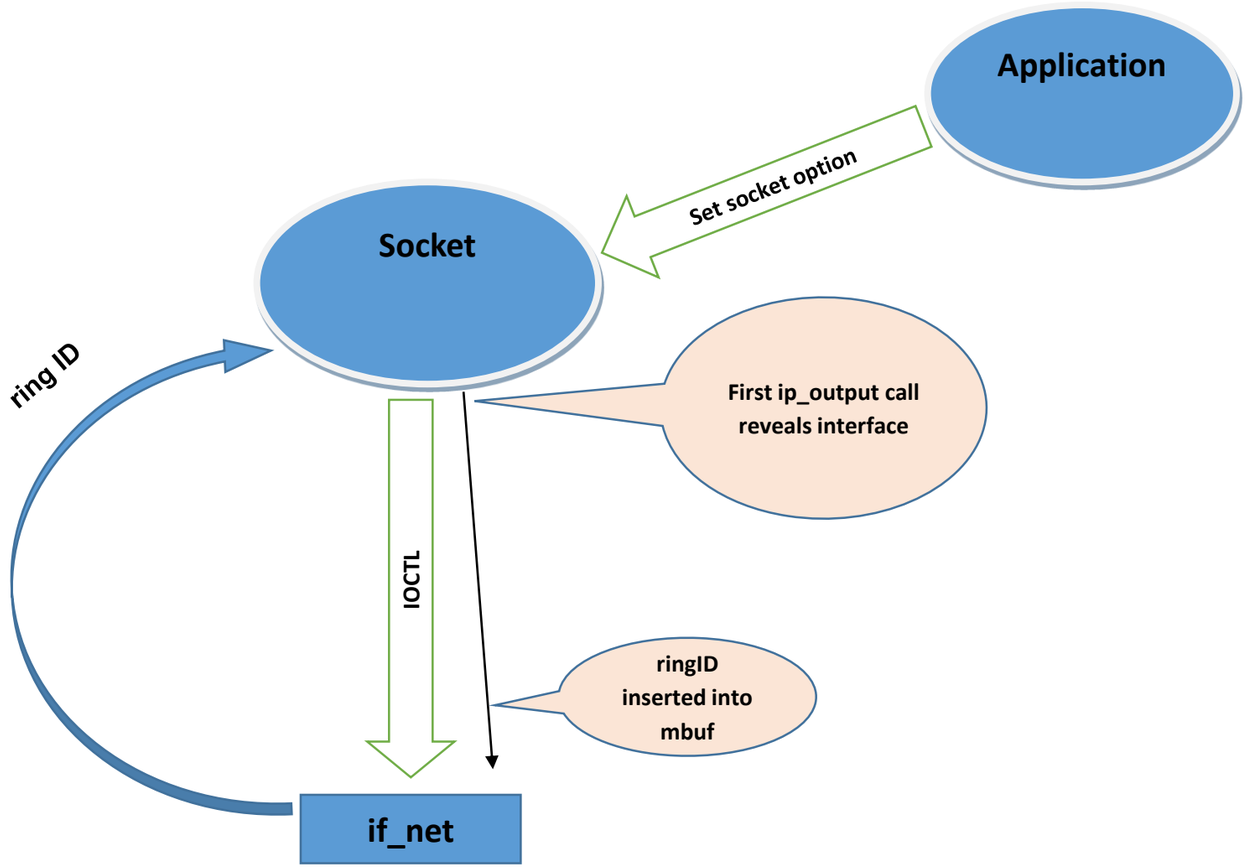
What is Packet Pacing?

- Rate limited TCP/UDP socket based connections

- Feature qualities:
  - Control Max bandwidth sent
  - Different rates for different flows
  - Smooth and even distribution between flows
  - Minimal bursts sent to the network
  - Avoid congestions in the network

- Mellanox solution:
  - Hardware based implementation
  - Reducing CPU overhead compared to software solutions

Architectural overview



User space and Kernel - Main Flow

- Application opens a socket and set max rate limit (RL) value using ‘SO_MAX_PACING_RATE’ socket option
- First ip_output() call:
  - reveals relevant interface
  - triggers an IOCTL to allocate a new ringID with the requested rate
- IOCTL returns with ringID assigned by driver.
- All transmit mbufs will contain this ringID.

- Upon if_net change:
  - IOCTL old if_net to free the ringID
  - IOCTL new if_net to allocate new ringID

- Connection termination triggers IOCTL to free the ringID.

Error flow

- Add rate failure → socket's max pacing rate set to 0
- Modify rate failure → same outcome, old rate not preserved

- User to query socket using getsockopt() to verify rate is valid

Implementation - Main changes

sys/socketvar.h:

```
struct socket {
+           uint32_t so_max_pacing_rate;
            }
```

sys/mbuf.h:

```
struct mbuf {
+           #define     M_HASHTYPE_TXRTLMT          11
            }
```

sys/netinet/in_pcb.h:

```
struct inpcb {
+           struct  ifnet *inp_txringid_ifp;
+           uint32_t inp_txringid_max_rate;
+           uint32_t inp_txringid;
            }
```

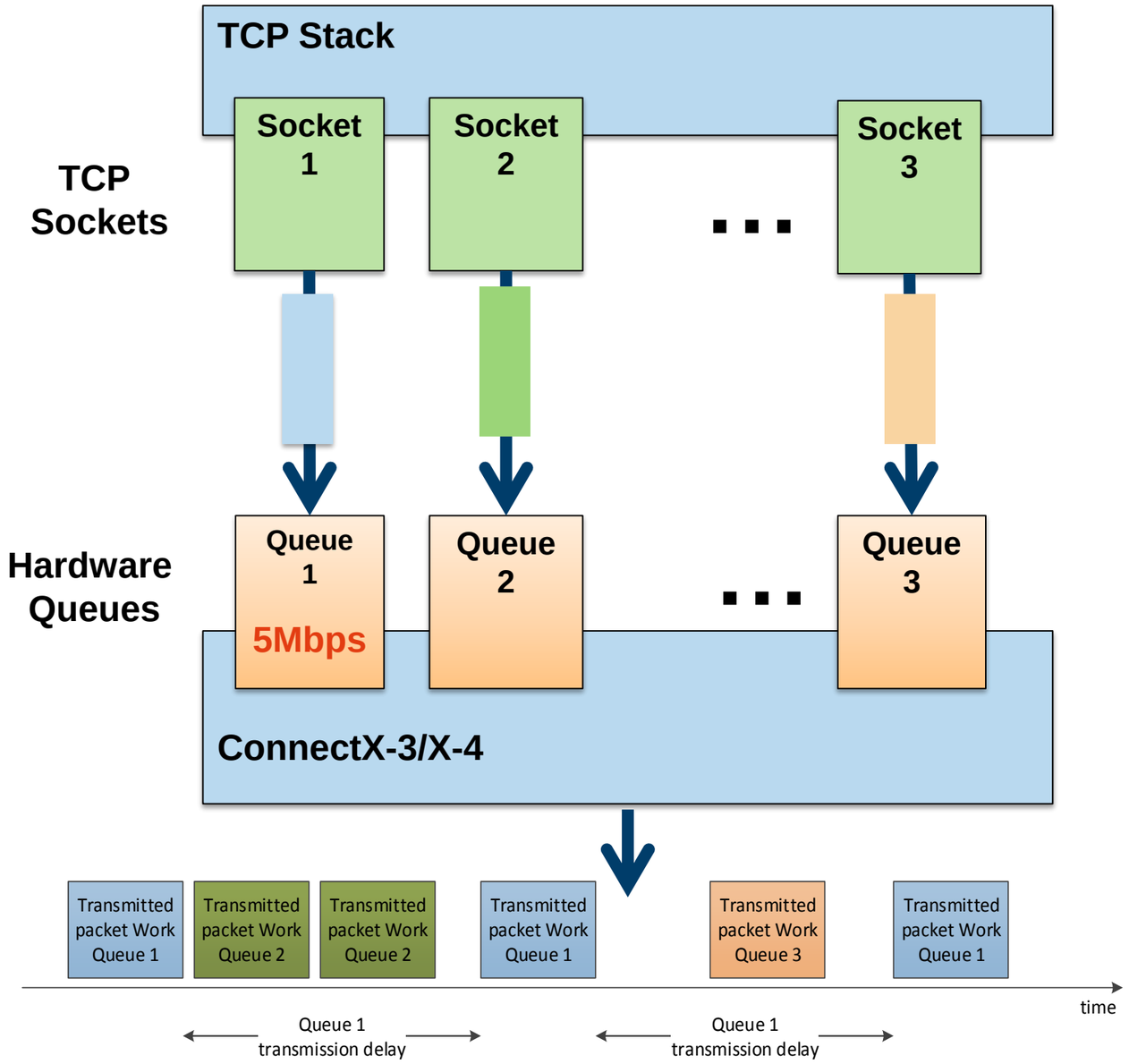
sys/sys/sockio.h:

```
+     #define     SIOCARATECTL      _IOWR('i', 139, struct ifreq_txrtlmt)
+     #define     SIOCSRATECTL      _IOWR('i', 140, struct ifreq_txrtlmt)
+     #define     SIOCDRATECTL       _IOW('i', 141, struct ifreq_txrtlmt)
```

Code overview

Mellanox driver - highlights

- Designated ring per rate limited connection
- The rate limiter can be configured with any value with a 1Mb/sec granularity
- IOCTL call from data path → Driver returns ringID immediately.
  Resource creation/destruction will happen asynchronously
- Manage resources while HW performs actual rate limiting
- Relevant capabilities will be communicated via sysctl
- Feature support via IFCAP_TXRTLMT



sysctl output

- sys.device.mlx4_core0.rate_limit_caps.max_value: 50 Mbps
- sys.device.mlx4_core0.rate_limit_caps.min_value: 250 Kbps

- sysctl hw.mlxen1.conf.rate_limit_show | head

  hw.mlxen1.conf.rate_limit_show:

INDEX	CURRENTLY USED	BURST	RATE [bit/s]
1	0	LOW	400,000
2	0	LOW	1,000,000
3	0	LOW	2,000,000
4	0	LOW	4,000,000
5	0	LOW	0
6	0	LOW	0

Characteristics – Mellanox ConnectX-3

- Number of rate limited connections - up to 45K
- Supported rates: 250Kb/s - 50Mb/s
- Road map: VLAN, LAGG
- Link to fabricator: https://reviews.freebsd.org/D3687

Demo

Questions?

Thank You!

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™