

# Re: Changes in the network interface queuing handoff model

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2006-08/msg00000.html>

---

- *From:* Andrew Gallatin <[gallatin@xxxxxxxxxxx](mailto:gallatin@xxxxxxxxxxx)>
  - *Date:* Tue, 1 Aug 2006 08:30:33 -0400 (EDT)
- 

Robert Watson writes:

5BOne of the ideas that I, Scott Long, and a few others have been bouncing around for some time is a restructuring of the network interface packet transmission API to reduce the number of locking operations and allow network device drivers increased control of the queuing behavior. Right now, it

<....>

– The ifnet send queue is a separately locked object from the device driver, meaning that for a single enqueue/dequeue pair, we pay an extra four lock operations (two for insert, two for remove) per packet.

Going forward, especially now that we support sun4v CoolThreads hardware, we're going to want to rethink the "single lock" per transmit routine model that most drivers have. The most expensive operation in transmit routines is `bus_dmamap_load_mbuf_sg()`, especially when there is an IOMMU involved (like on CoolThreads machines) and there is no reason why this needs to be called with a driver's transmit lock held. I have hard data (from Solaris) about how much fine grained locking in a 10GbE driver's transmit routine helps.

Drew

---

[freebsd-arch@xxxxxxxxxxx](mailto:freebsd-arch@xxxxxxxxxxx) mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "[freebsd-arch-unsubscribe@xxxxxxxxxxx](mailto:freebsd-arch-unsubscribe@xxxxxxxxxxx)"