# Network Paradigms

Standard Linux networking VS Intel DPDK

Neel Shah

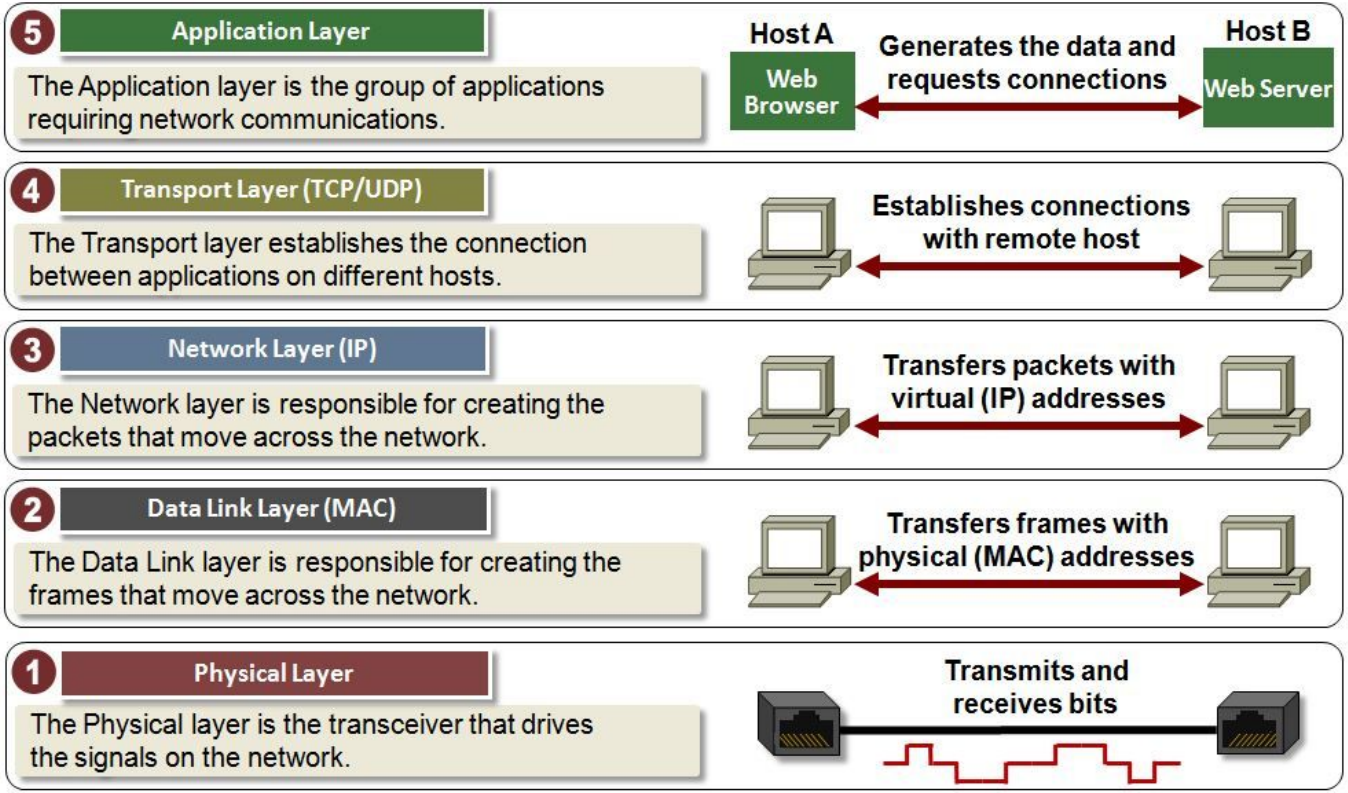
# Outline

- Networking in Linux review
- Network device architecture
- Dance of Interrupts
- Cool Developments
- Intro to DPDK
- Performance: standard networking VS DPDK

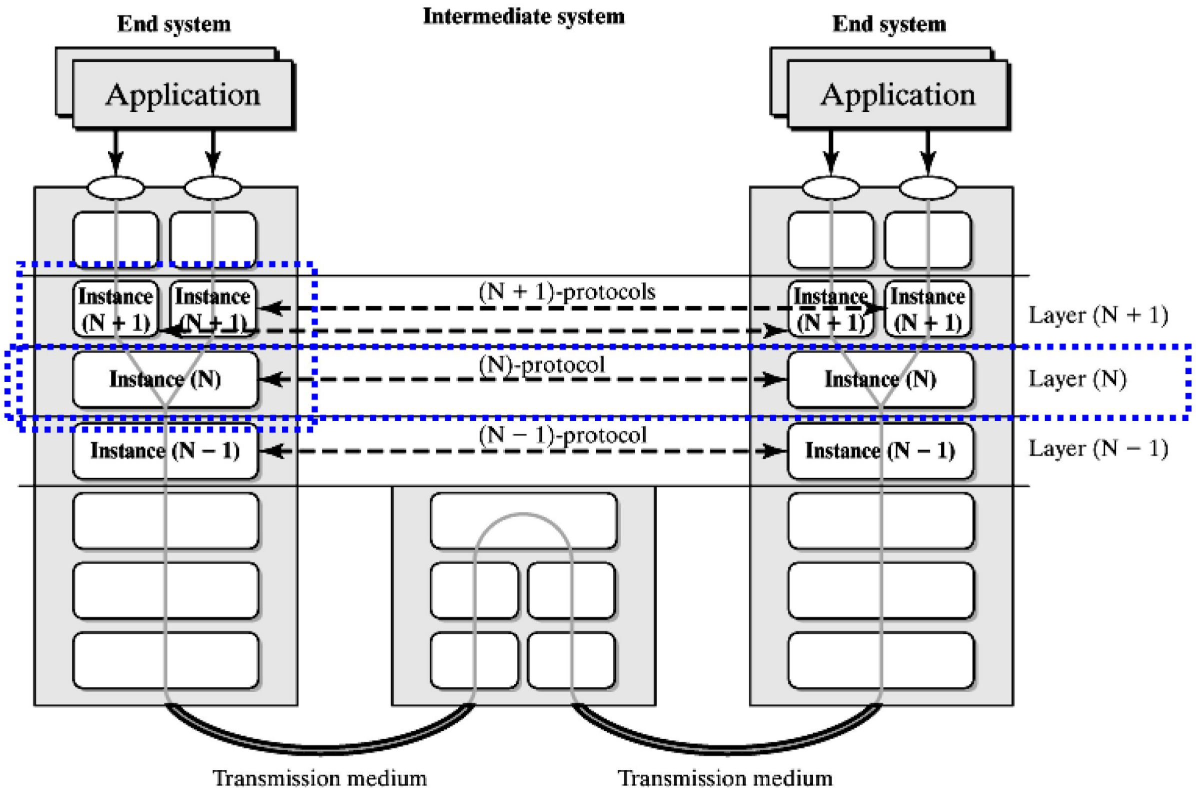
# Outline

- Networking in Linux review
- Network device architecture
- Dance of Interrupts
- Cool Developments
- Intro to DPDK
- Performance: standard networking VS DPDK

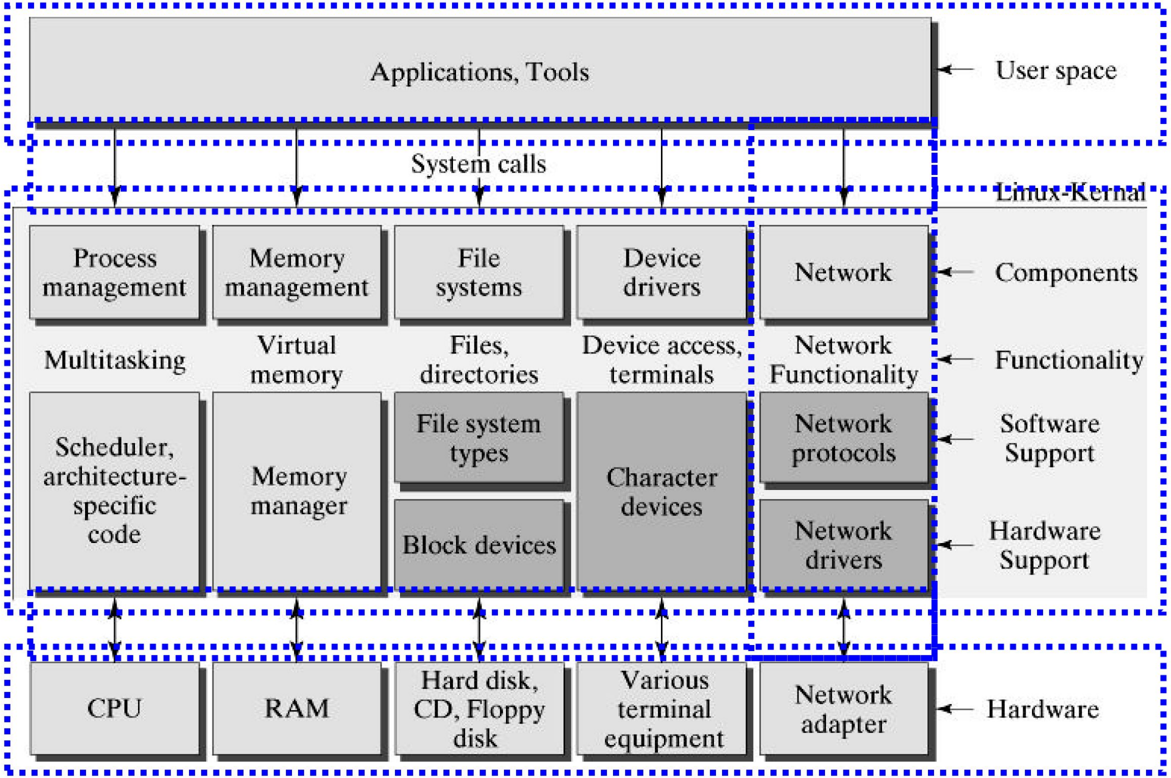
# OSI/TCP Stack



# OSI/TCP Stack Cont.

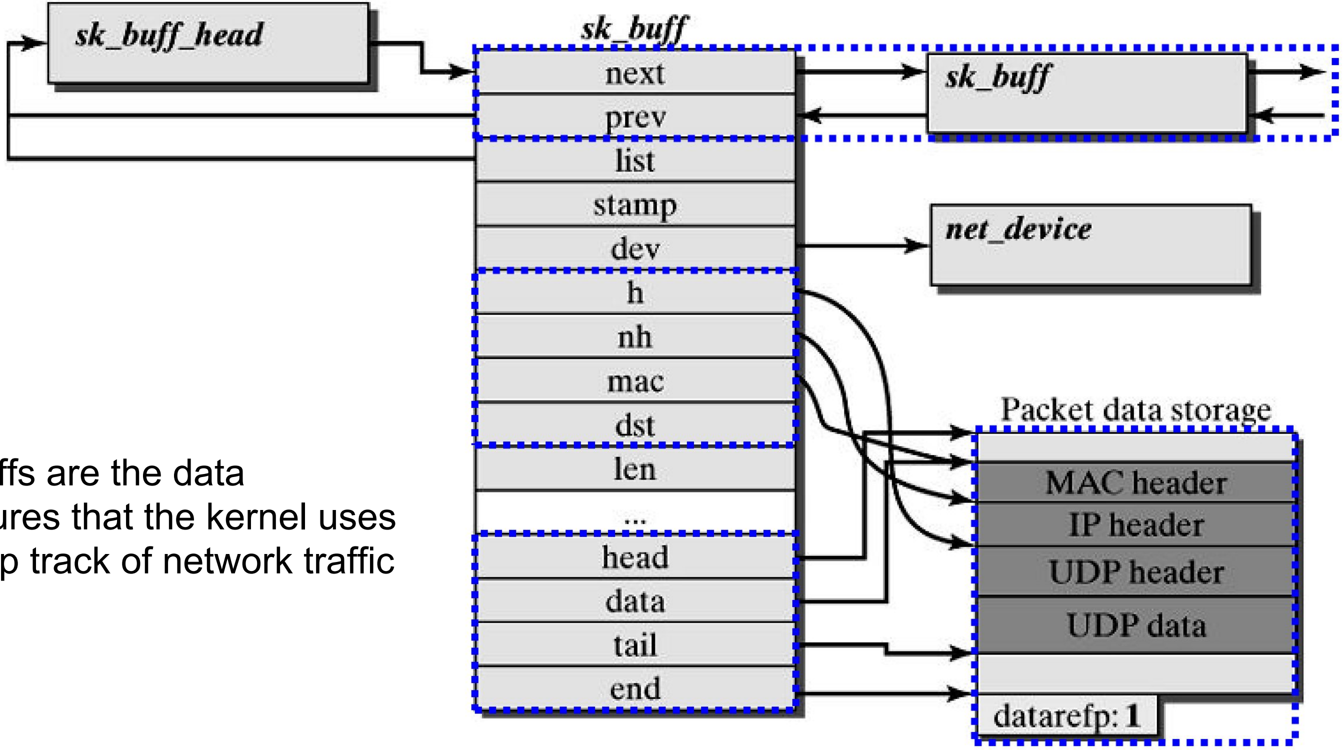


# Linux Perspective




[Interactive Linux Kernel Map]

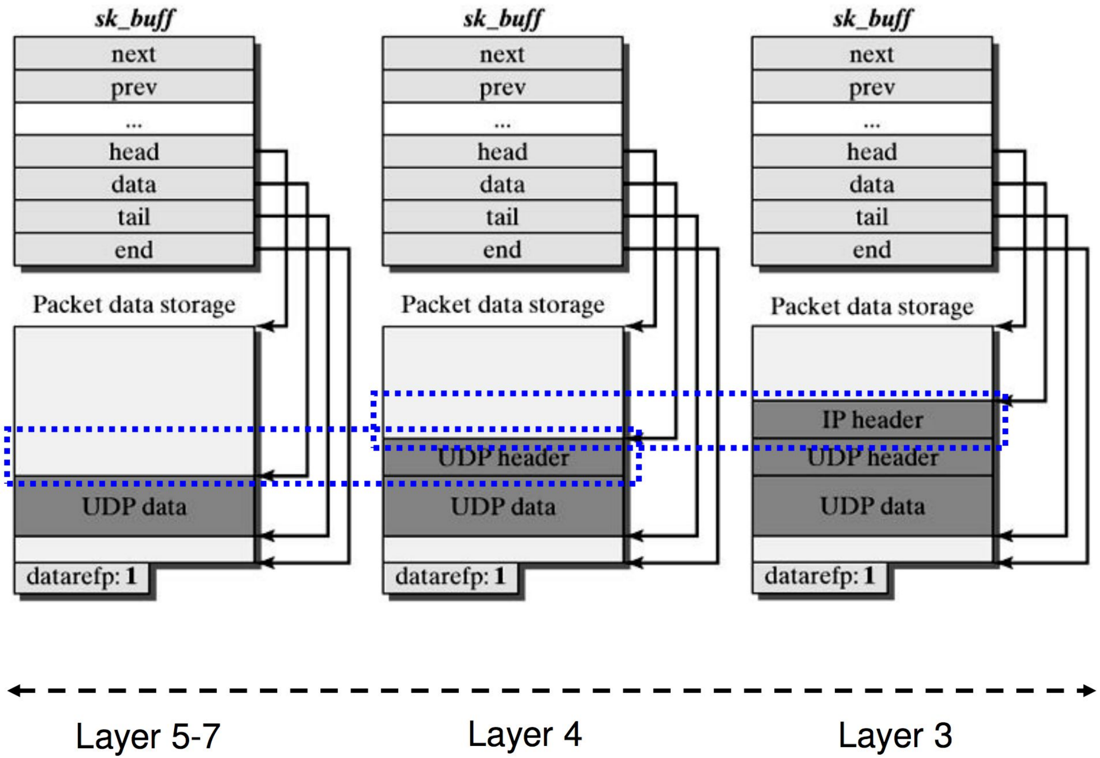
# Socket Buffer (sk_buff)




- sk_buffs are the data structures that the kernel uses to keep track of network traffic

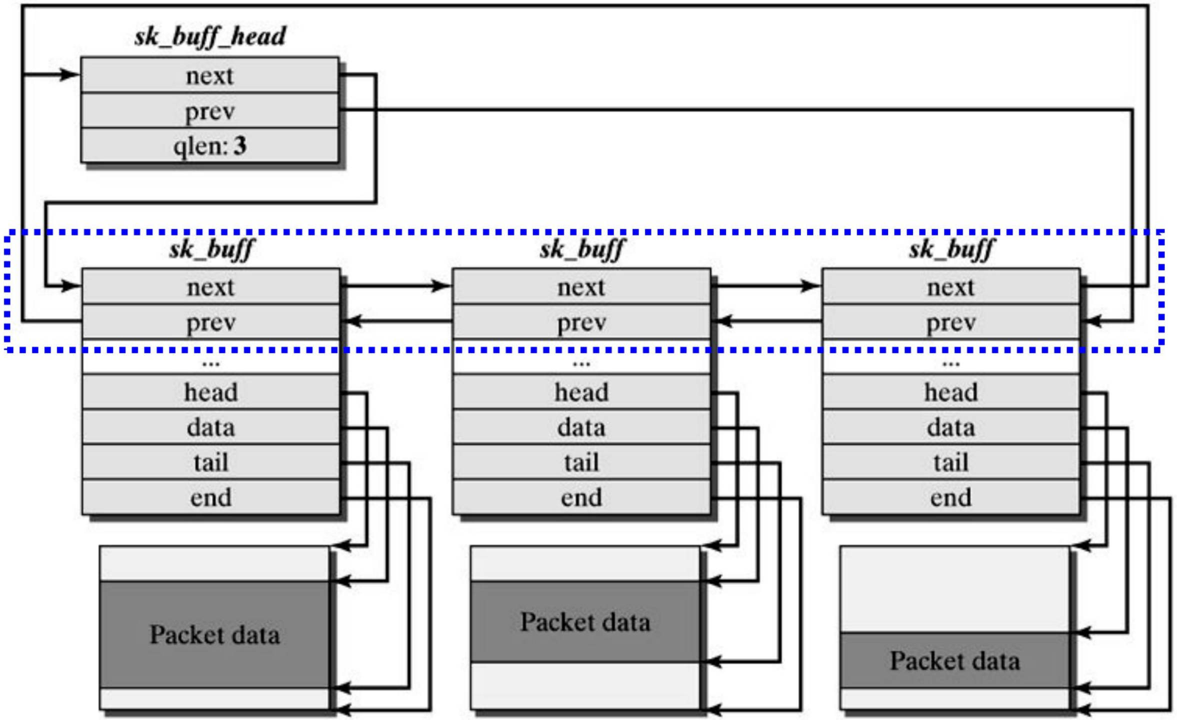
# sk_buffs cont.

- Each layer has a callback function to process the packet
- The callback function stores that layer's header into the sk_buff



# sk_buffs cont.

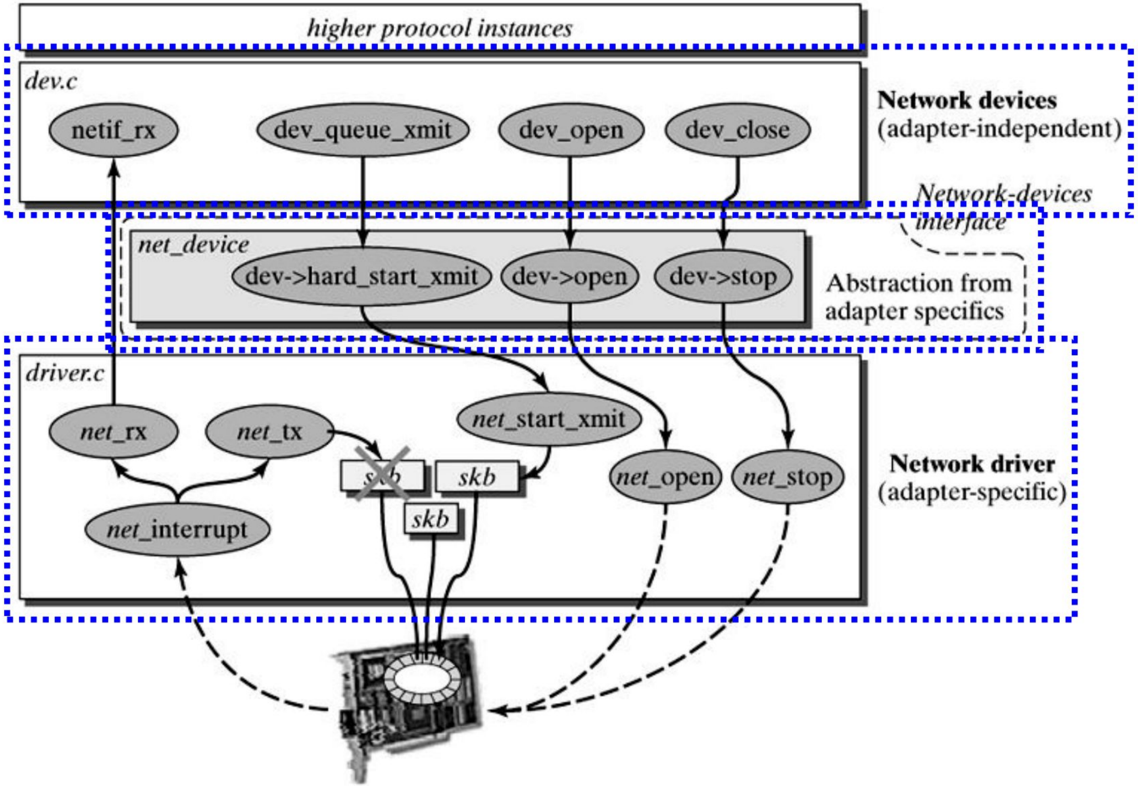
- Kernel keeps a queue of RX and TX sk_buffs
- Hardware pre-allocates sk_buffs
- Memory addresses configured for DMA



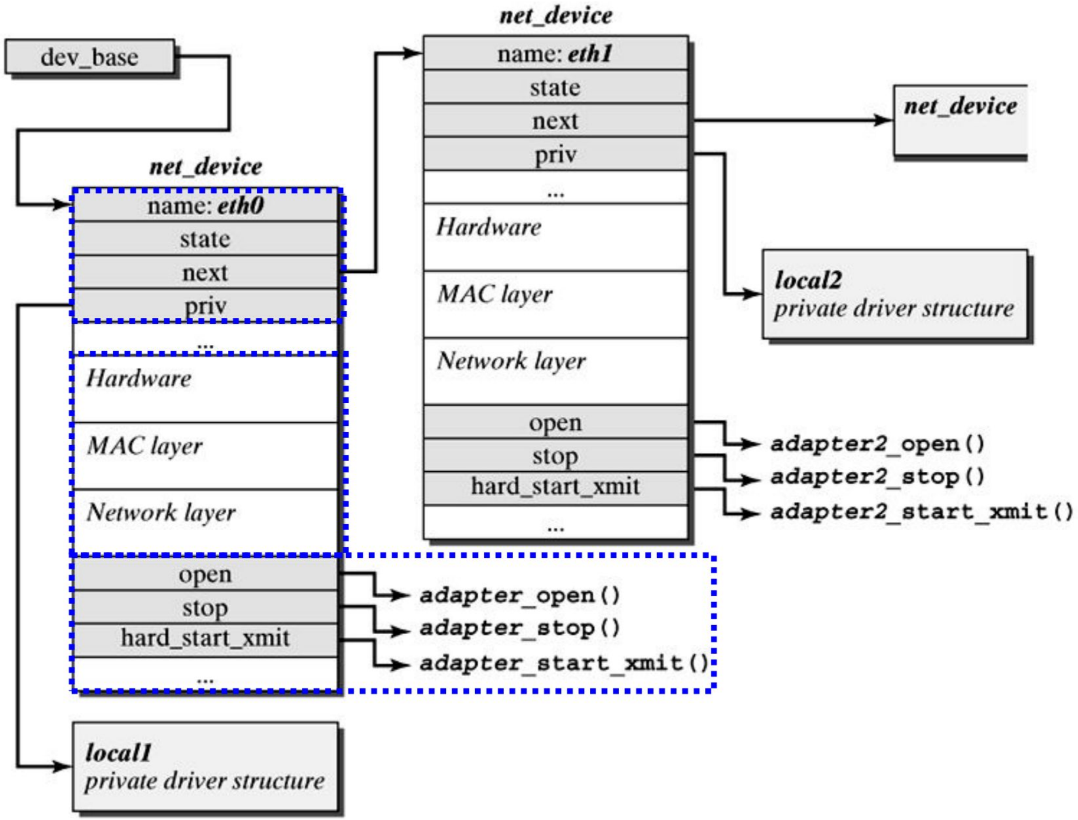
# Outline

- Networking in Linux review
- Network device architecture
- Dance of Interrupts
- Cool Developments
- Intro to DPDK
- Performance: standard networking VS DPDK

# Device Interface Overview

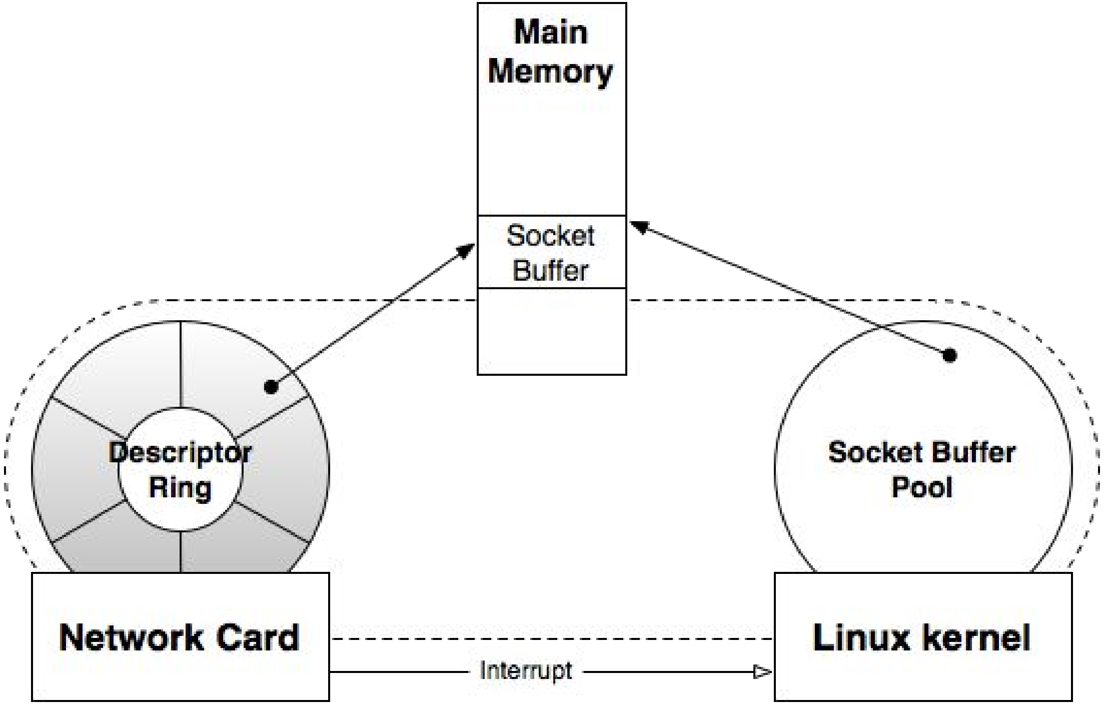


# Device Interface Overview



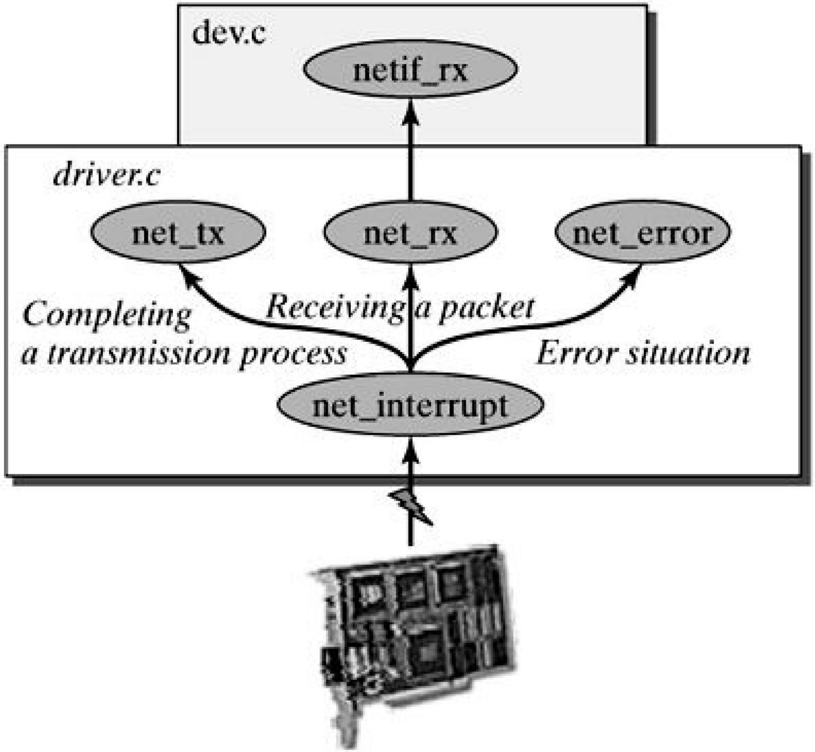
# Device Initialization

- Driver loaded as module upon boot
- Register driver with PCI bus and enable the device on the bus (`igb_init_module`)
- Open (`igb_open`) the network device and allocate all rx and tx resources and setup the interrupt handler
    - TX first, RX second
- Create wrapper buffer to link hardware descriptor to software socket buffer pool (`igb_configure`)



# Network Device Interrupts

- Two interrupt routines
  - hardware interrupt routine
  - software interrupt routine
- RX queue on network device requests interrupt number and sets the interrupt routine (`igb_request_irq`)
- Register softIRQ with NAPI layer (`netif_napi_add`)

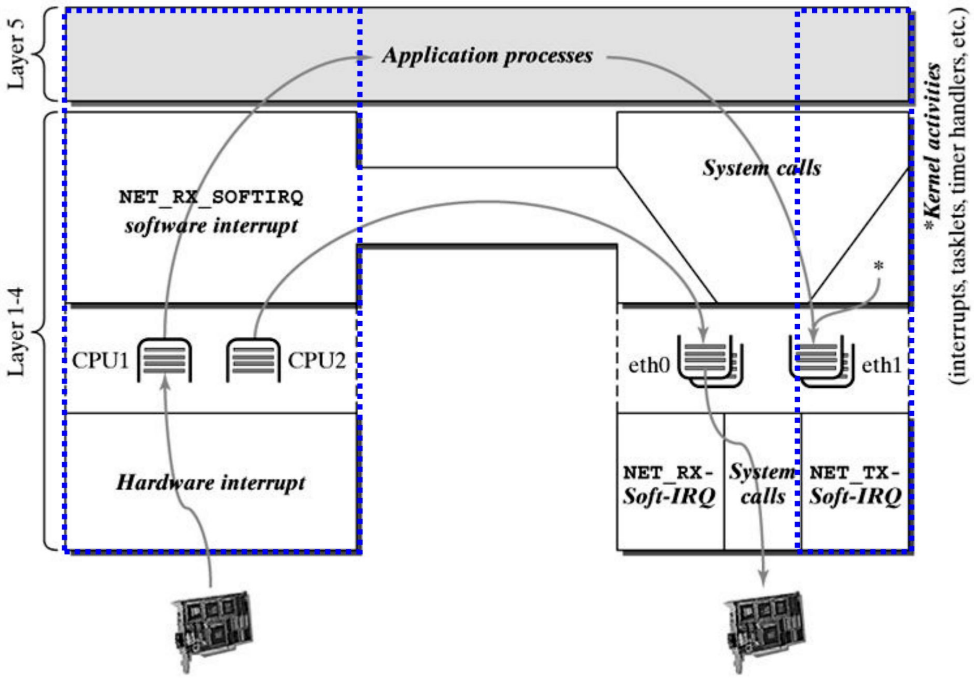


# Outline

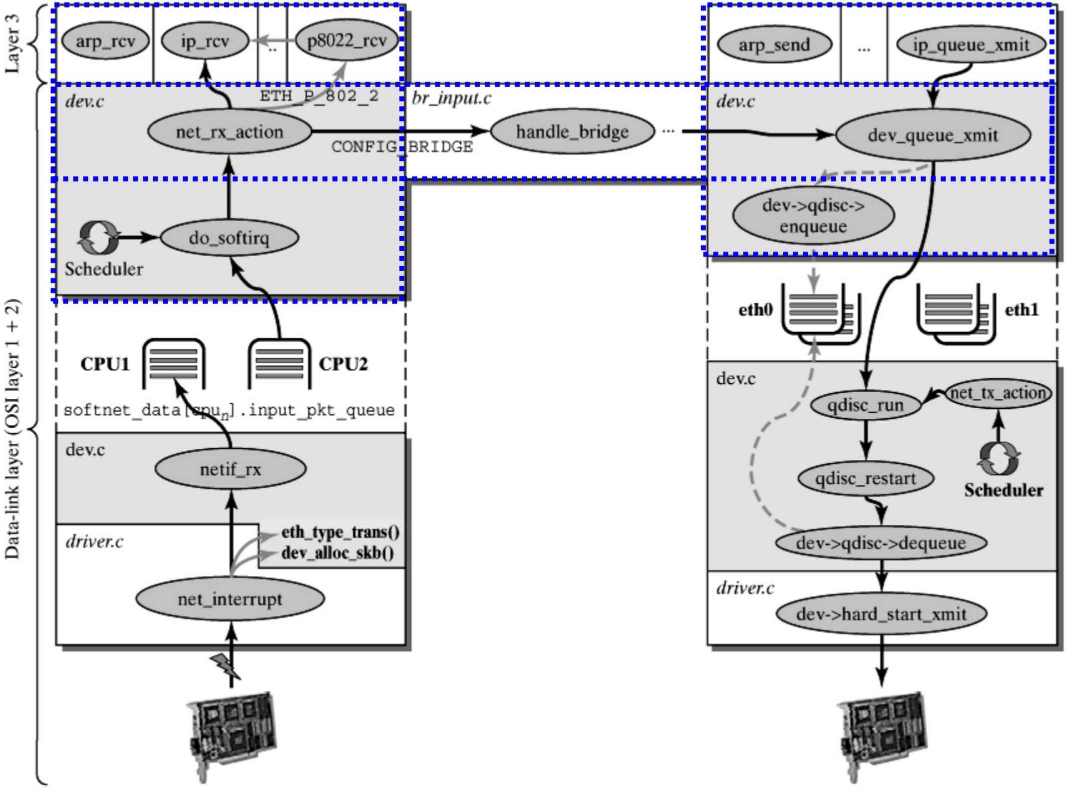
- Networking in Linux review
- Network device architecture
- Dance of Interrupts
- Cool Developments
- Intro to DPDK
- Performance: standard networking VS DPDK

# RX Packet

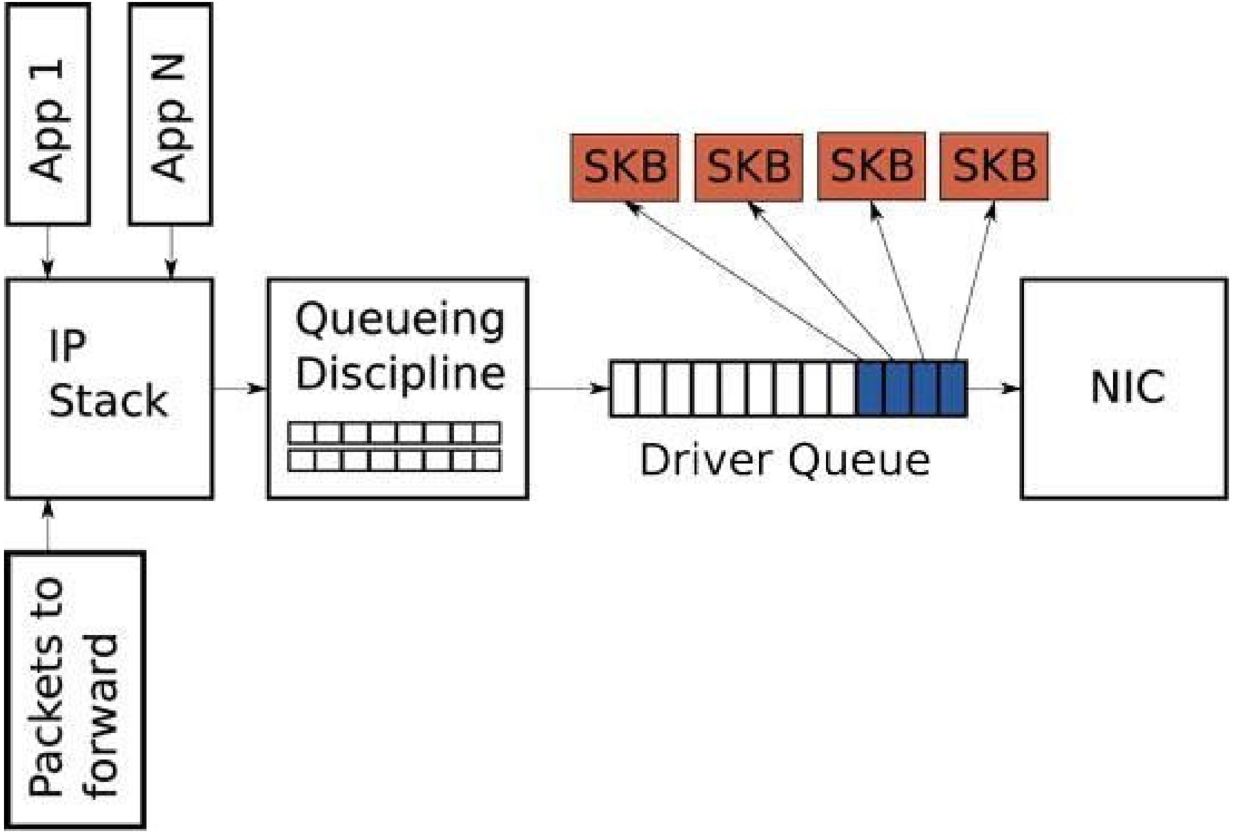
- When packet first arrives:
  - Device writes packet to next HW descriptor (location of socket buffer pool)
  - Once fully received, assert interrupt
- HW Interrupt
  - Find CPU associated to RX ring and place packet there
  - Signals softIRQ
  - Un-assert HW interrupt
- softIRQ
  - Schedule softIRQ handler
  - When application reads from socket, give packet



# RX/TX Process



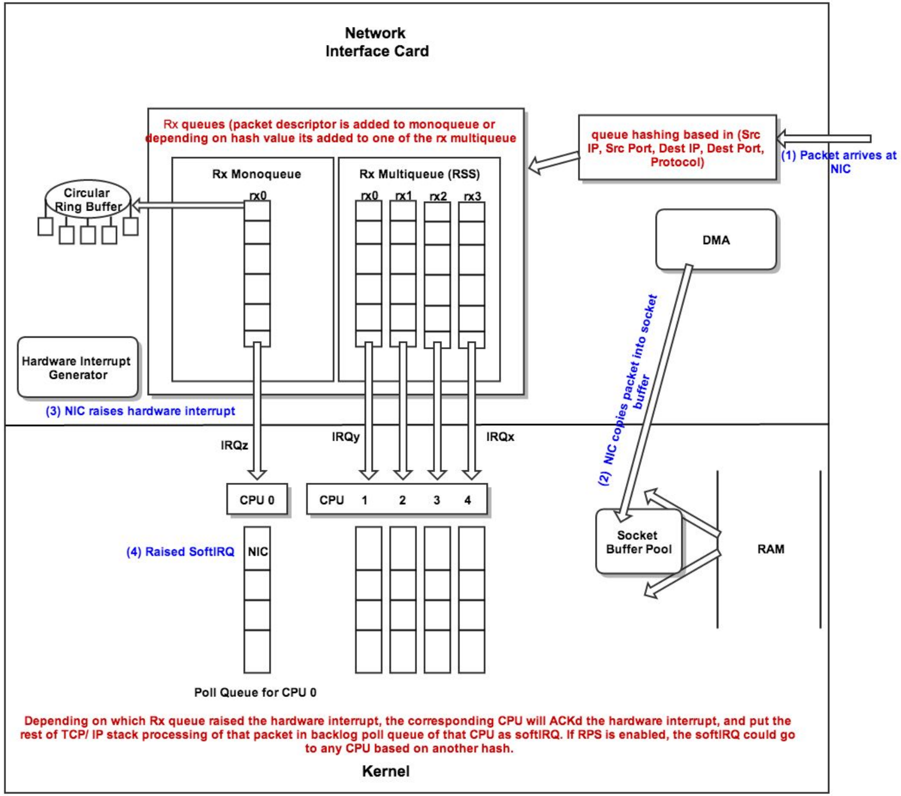
# RX/TX Process



# Outline

# Receive Side Scaling (RSS)

- Before, each NetDev had one RX queue
- Modern NetDev's can have many RX queues
- Each RX queue is associated to different processor
    - Uniform load distribution across processors
- NetDev multiplexes RX traffic to determine which queue gets the packet
- Now, hardware interrupt hits different processor based on RX queue



# Message Signaled Interrupts (MSI-X)

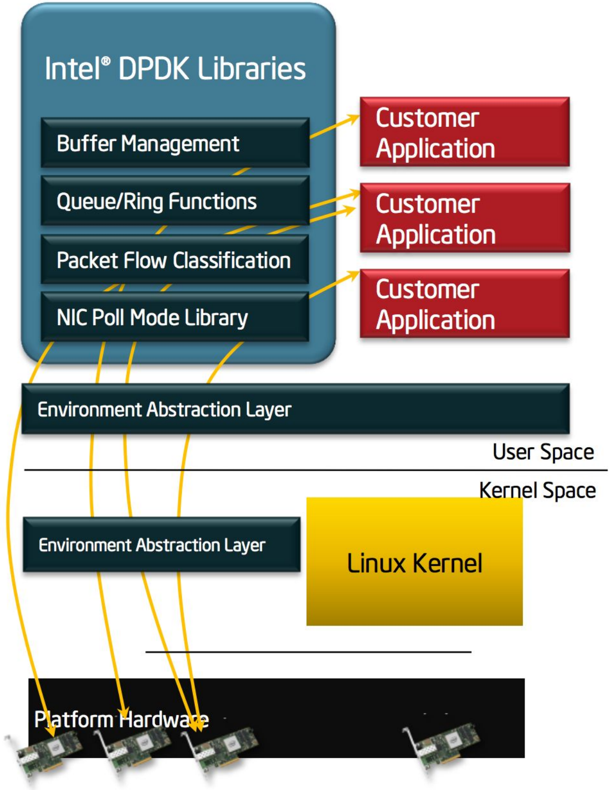
- Allows more interrupts than there are physical pins for interrupts
    - Replaces dedicated interrupt lines by allowing device to write interrupt describing data to special mmapped I/O addresses
    - Chipset then delivers interrupt to corresponding processor
    - MSI-X permits PCI 3.0 >= devices to allocate up to 2048 interrupts
- Unique interrupt for each receive queue
    - Kernel knows what caused each interrupt
    - User able to pin interrupts from specific queue to specific processor queue to minimize cache effects (no likey)

# Outline

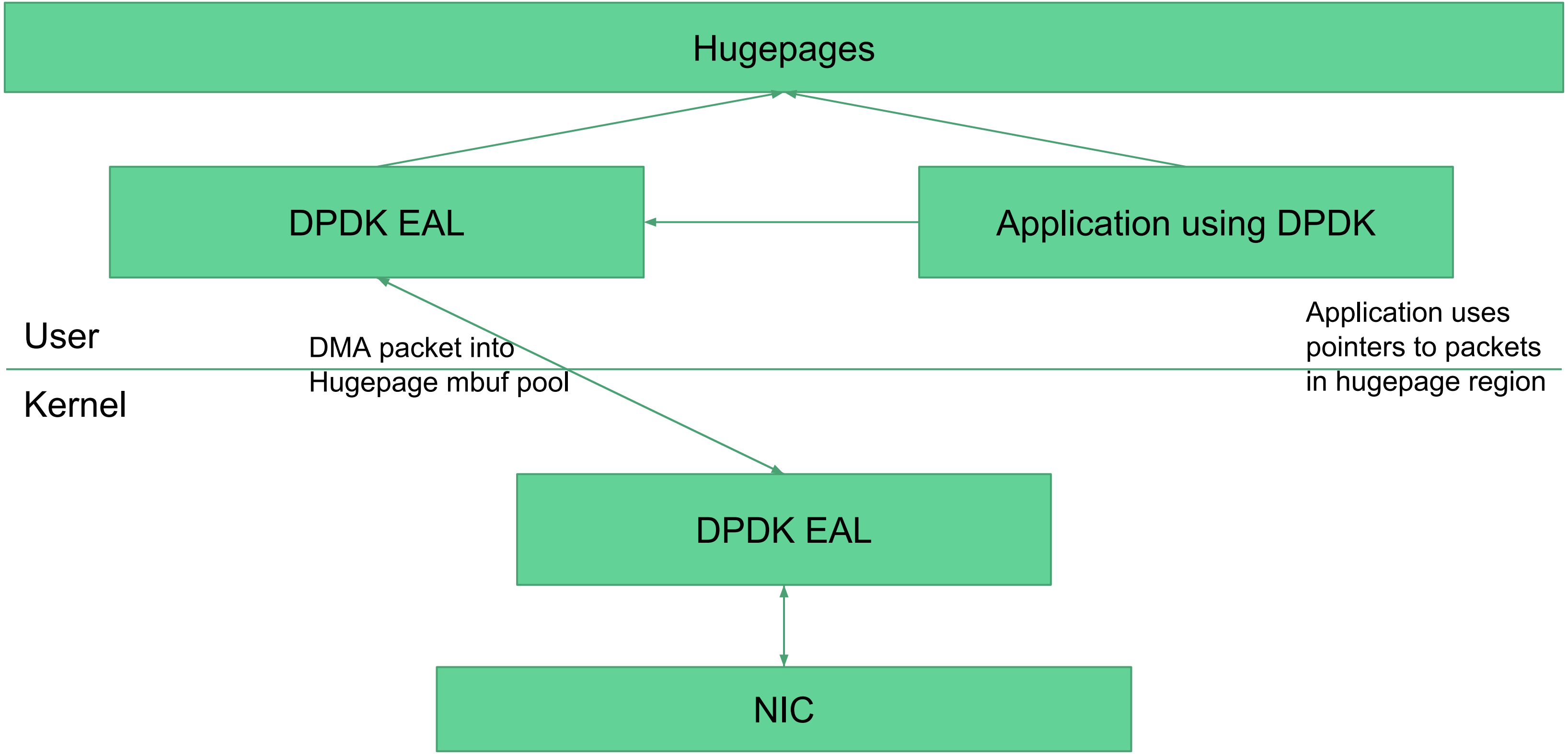
- Networking in Linux review
- Network device architecture
- Dance of Interrupts
- Cool Developments
- Intro to DPDK
- Performance: standard networking VS DPDK

# Intro to Intel Data Plane Development Kit (DPDK)

- Memory Manager
  - Hugepage memory space
  - Allocates pools in hugepage memory
  - Alignment helper to distribute and pad objects across DRAM channels
- Buffer Manager
  - allocate/deallocate fixed size buffers stored in memory pools
- Queue Manager
  - Lockless queues which allow various software components to process packets
- Poll Mode Drivers
  - Uses polling instead of asynchronous interrupt-based signalling to speed packet pipeline



# DPDK Architecture



# DPDK Poll Mode Driver and Details

- Recap DPDK:
  - RX traffic DMA’d into Hugepage region
  - Packet data stripped of standard packet headers and wrapped in mbuf structure
  - Applications using DPDK request a burst of packets to process
  - Send via burst when done
  - Poll mode driver sends packets out of NIC avoiding kernel
- NIC and kernel must be able to use RSS
- NetDevice has many RX/TX queues to achieve line speeds of 40+ Gbps

# Outline

- Networking in Linux review
- Network device architecture
- Dance of Interrupts
- Cool Developments
- Intro to DPDK
- Performance: standard networking VS DPDK

# Performance: Standard Net VS DPDK

- In standard networking, where are the context switches?
- How does RSS and MSI-X impact network processing?
- How does DPDK speed things up on a Linux level?
- How can we take these interrupt handling models to other devices to improve performance?

# Credits/Further Reading


- http://www.slideshare.net/hugolu/the-linux-networking-architecture
- http://beyond-syntax.com/blog/2011/03/diving-into-linux-networking-i/
- https://www.kernel.org/doc/Documentation/networking/scaling.txt
- http://balodeamit.blogspot.com/2013/10/receive-side-scaling-and-receive-packet.html
- https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/network-rss.html
- https://www.kernel.org/doc/Documentation/PCI/MSI-HOWTO.txt
- https://en.wikipedia.org/wiki/Advanced_Programmable_Interrupt_Controller
- https://en.wikipedia.org/wiki/Message_Signaled_Interrupts
- http://www.intel.com/content/dam/www/public/us/en/documents/presentation/dpdk-packet-processing-ia-overview-presentation.pdf