pps Matters

Muhammad Moinur Rahman
moin@bofh.im

What is a switch/router?

- A switch forwards frame based on MAC address
- A router forwards packets based on IP address

What is a Software Switch/Router?

- Software based implementations
- Routers
  - BIRD, FRR, Zebra, Quagga, ExaBGP
- Switches
    - Open vSwitch
- Mostly installable in a Virtualized Environment or on a *nix environment

What is Hardware Switch/Router?

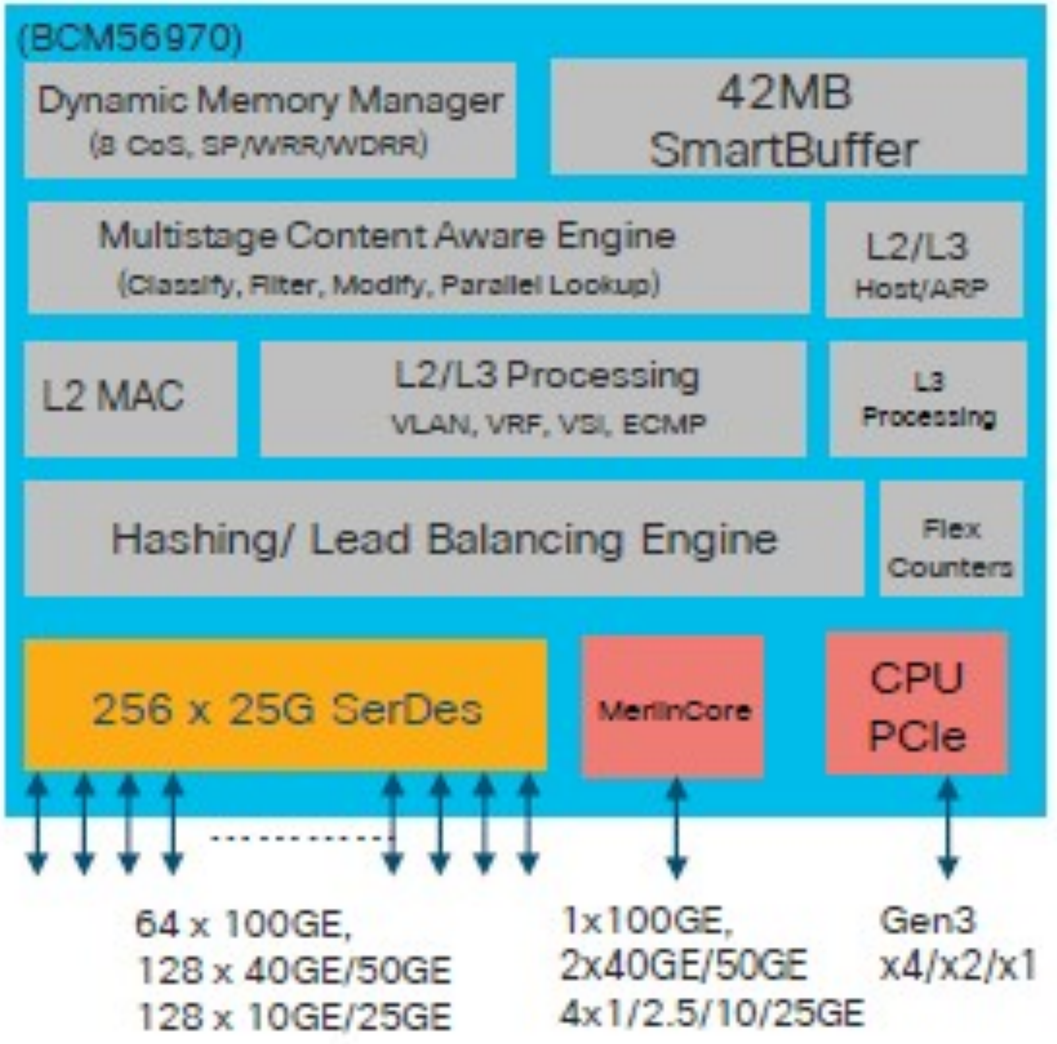
- Manufactured by big names like Cisco, Juniper, ARISTA, Extreme, Nokia
- Comes with Price Tag
- Sometime comes with really big size
- Has different and multiple ports
  - * X 1/10/25/40/50/100/400GB
- So many jargons
  - ASIC/Merchant Silicon
  - GBPS/TBPS backplane capacity
  - GBPS/TBPS forwarding capacity
  - k/K/m/M pps forwarding
  - line rate forwarding

What is ASIC/merchant Silicon?

- ASIC Miners - Just one example
- Application Specific Integrated Circuits
- Some applications
  - Bitcoin Miner
  - Voice Recorder
  - Cryptographic Accelerator
  - Network Switches
  - Firewalls
- New Lingo for DC Switches is Silicon
- Off the shelf or Custom Built ASICs
- Broadcom, Cavium are some Silicon Manufacturers
- Broadcom Tomahawk is the flagship ASIC

Tomahawk2 ASIC Architecture

- BCM56970 from StrataXGS family
- 6.4Tbps Single Chip Ethernet Switch
- 4 Pipes @1.6 Tbps
- 42MB (4x10.5MB) of Buffer
- 64 Falcon Cores
- 1 Merlin Core
- Ingress & Egress Packet Time Stamping



The BIG Questions

1. If there are open source switch/routers why do we need to buy price tagged Vendor Devices?
2. Why use Silicon or chips instead of generic X86 processors
3. *nix OS can do anything. Why don’t we install those apps and get rid of Hardware Vendors?

x86 vs ASIC

- x86
  - Jack of all, master of none
  - CPU and PCI interrupts
  - Limited PCIe bandwidth and based on CPU arch
- ASIC
  - Master of one
  - No interrupts
  - Sky is the limit for PCIe bandwidth

POSIX poses

- POSIX sockets evolved from Berkley Sockets
- BSD Sockets are still the defacto standard since 4.2 BSD Unix
- Adopted from Linux to Windows
- Basic life cycle
  - socket(), bind(), listen(), accept(), sendmsg(), recvmsg()
- Network Stacks are implemented in-kernel
- So the functions are using system-call
- Higher overhead for Context Switch and CPU Cache Pollution
- Back-and-forth game in Multi-Core CPU and Multi Queue NIC
- socket buffers(skb) or network memory buffer(mbuf) stresses OS memory allocators

Mind the GAP

- Minimal pause required between packets or frames
- Interpacket GAP/Interframe spacing/Interframe GAP
- The standard is 96 bit times
- 9.6 μs for 10 Mbit/s Ethernet
- 0.96 μs for 100 Mbit/s (Fast) Ethernet
- 96 ns for Gigabit Ethernet
- 38.4 ns for 2.5 Gigabit Ethernet
- 19.2 ns for 5 Gigabit Ethernet
- 9.6 ns for 10 Gigabit Ethernet
- 2.4 ns for 40 Gigabit Ethernet
- 0.96 ns for 100 Gigabit Ethernet

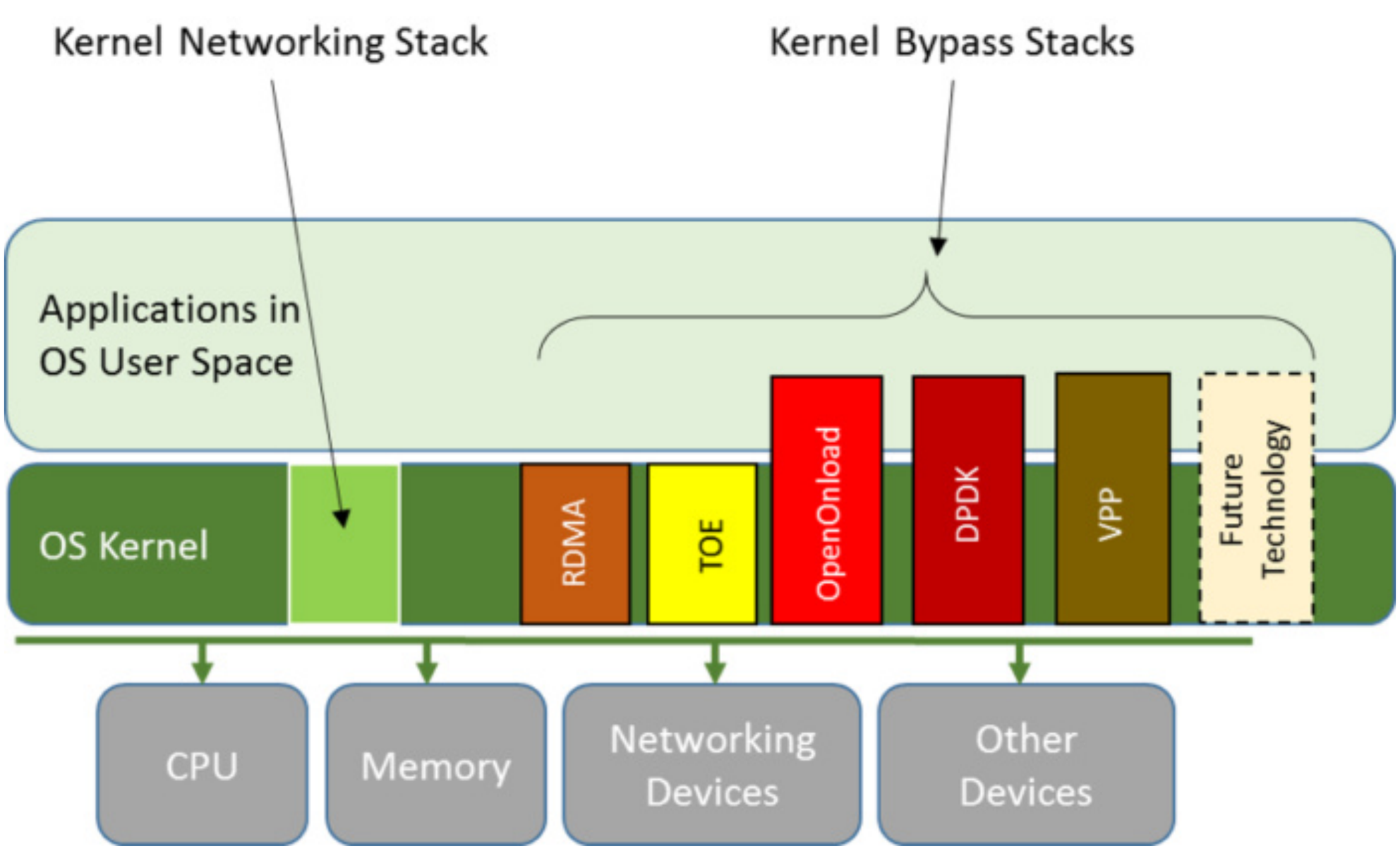
run KERNEL run

- KERNEL processing time for 1538 bytes of frame
  - at 10Gbit/s == 1230.4 ns between packets (815Kpps)
  - at 40Gbit/s == 307.6 ns between packets (3.26Mpps)
  - at 100Gbit/s == 123.0 ns between packets (8.15Mpps)
- Smallest frame size of 84 bytes
  - at 10Gbit/s == 67.2 ns between packets (14.88Mpps)
- CPU budget
  - 67.2ns => 201 cycles (@3GHz)

OS Limitation

- Most OS are jack of all and master of none
- Desktop, Mail Server, Web Server, DNS Server
- Graphics Rendering, Gaming, Day to Day work
- They are not designed for performance packet processing
- Not optimized for line rate packet processing
- Vyatta, bsdrp are to name a few
- Lots of other commercial os

- That is not the END GAME



kernel bypass

zero-copy

- CPU skips task of copying Data from one memory area to another
  - Saves CPU cycles
  - Saves memory bandwidth
- OS elements
  - Device Driver
  - File Systems
  - Network Protocol Stack
- zero-copy versions
- Reduces number of mode switching between kernel space and user space applications
- mostly uses raw sockets with mmap(Memory Map)
- kernel bypass utilizes zero-copy and they arre not the same

RDMA

- Remote Direct Memory Access
  - Implemented over high speed, low-latency networks(fabrics)
  - Direct access to remote host’s memory
  - Dramatically reduces latency and CPU overhead
- Requires specialized hardware specially NIC with support for RDMA
- Bypass remote or local operating system
- Transfers data in between wire and application memory
- Bypasses CPU, cache and context switching
- Transfer continues parallel with OS operations without affecting OS performance
- Applications can or cannot be RDMA aware

RDMA(continued)

- Link Layer protocol can be
  - Ethernet
  - iWARP(internet Wide Area RDMA Protocol) combines with TCP Offload Engine
    - NVMe over Fabrics(NVMEoF)
    - iSCSI Extensions over RDMA(iSER)
    - SMB Direct
    - Sockets Direct Protocol(SDP)
    - SCSI RDMA Protocol(SRP)
    - NFS over RDMA
    - GPUDirect
- Link Layer protocol can be
  - InfiniBand
    - Oldest RDMA implementations
    - Main manufacturers were Intel and Mellanox
    - Mostly used in Super Computing environment
    - Ethernet can be run over InfiniBand
  - Omni-Path
    - Low Latency Networking Architecture by Intel

RoCE

- RDMA over Converged Ethernet
- Two versions
  - RoCEv1 focuses on Ethernet Link Layer mainly Ethertype 0x8915
  - RoCEv2 focuses on Internet Layer mainly UDP/IPv4 and UDP/IPv6
    - Routable RoCE is the other lingo of v2 due to it’s routable capability
- Also runs over non-converged Ethernet
- RoCE vs InfiniBand
  - RoCE requires lossless Ethernet
- RoCE vs iWARP
  - RoCE performs RDMA over Ethernet/UDP whereas iWARP uses TCP
- Some of the vendors are
  - Nvidia -> Mellanox
  - Broadcom -> Emulex
  - Cavium -> QLogic/Marvel Technology

The Cool People of Internet

- Connection Establishment (SYN;SYN-ACK;ACK)
- Acknowledgement of traffic receipt
- Checksum and Sequence
- Sliding Window Calculation
- Congestion Control
- Connection Termination

TOE(TCP Offload Engine)

- Offloads kernel TCP stacks in NIC
- Free up host CPU cycles
- Reduces PCI traffic in between PCI bus and host CPU
- Types
  - Parallel-Stack Full Offload
    - Host OS TCP/IP stack and parallel stack with “vampire tap”
  - HBA full Offload
    - Host Bus Adapter used mainly in iSCSI host adapters
    - Besides TCP it also offloads iSCSI functions
  - TCP chimney partial Offload
    - Mainly a Microsoft lingo; but mostly used alternatively
    - Selective TCP stacks are offloaded

tso/lro

- TCP Segmentation Offload
  - Big chunks of data are split into multiple packets by NIC before transmission
  - The size depends on MTU of a link in between networking devices
  - NIC calculates and splits the data when offloaded from host OS
- Large Receive Offload
  - Just the opposite
  - Multiple packets of single stream are aggregated into single buffer before handing over to host OS reducing CPU cycle

chksum

- Although a weak check compared to modern checksum methods but TCP needs error checking
- Uses one’s complement algorithm
- This is CPU intensive work
- But can be offloaded into NIC if supported
- And it has some disadvantages:
  - If used along with packet analyzers; it will report invalid checksums for packets received
  - If used with some virtualization platform which do not have checksum offload capacity in it’s virtual nic driver

eco systems for fast packet processing

- There are lots of framework
- From open source to commercial
- Sometimes tightly coupled with a vendor
- Specially Network Interface Card vendor
- But there are open standards too
- Some eco systems are vnf friendly or offers application development API for building new solutions
- Commercial ones are really costly considering the price of NIC

xdp (eXpress Data Path)

- In Linux Kernel since 4.8
- eBPF based high performance Data path
- Similar to AF_PACKET a new address family AF_XDP
- Only supported in Intel and Mellanox cards
- eBPF is offloaded to NIC; in case drivers are unavailable then this is CPU processed and performs slower
- 26 Mpps per core drop test has been checked successfully with commodity hardware
- Designed for programmability
- This is not kernel bypass but rather integrated fast-path in kernel
- Works seamlessly with kernel TCP stack

pf_ring

- Available for Linux kernels 2.6.32 and newer
- Loadable kernel module
- 10 Gbit Hardware Packet Filtering using commodity network adapters
- Device driver independent
- Libpcap support for seamless integration with existing pcap-based applications.
- ZC version requires commercial license per mac
- User-space ZC (new generation DNA, Direct NIC Access) drivers for extreme packet capture/transmission speed as the NIC NPU (Network Process Unit) is pushing/getting packets to/from userland without any kernel intervention. Using the 10Gbit ZC driver you can send/received at wire-speed at any packet sizes.
- PF_RING ZC library for distributing packets in zero-copy across threads, applications, Virtual Machines.
- Support of Accolade, Exablaze, Endace, Fiberblaze, Inveatech, Mellanox, Myricom/CSPI, Napatech, Netcope and Intel (ZC) network adapters
- Kernel-based packet capture and sampling
- Ability to specify hundred of header filters in addition to BPF
- Content inspection, so that only packets matching the payload filter are passed
- PF_RING™ plugins for advanced packet parsing and content filtering
- Works pretty well within ntop ecosystem

DPDK(Data Plane Development Kit)

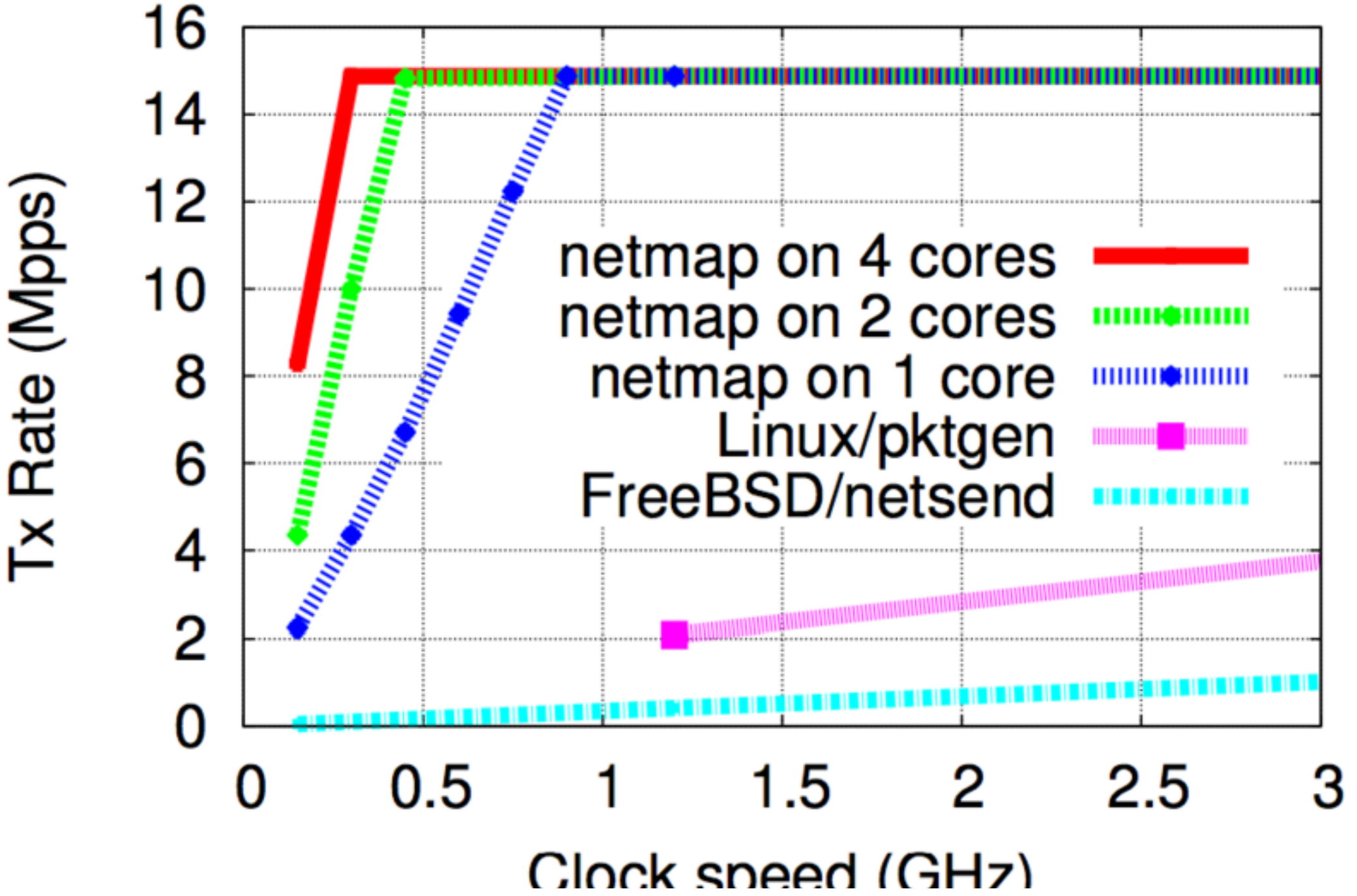
- Set of Data Plane libraries and NIC drivers
- Maintained by Linux Foundation but BSD licensed
- Programming framework for x86, ARM and powerPC
- Environment Abstraction Layer(EAL) is created consisting of a set of hardware/software environment
- Supports lots of hardware
  - AMD, Amazon, Aquantia, Atomic Rules, Broadcom, Cavium, Chelsio, Cisco, Intel, Marvell, Mellanox, NXP, Netcope, Solarflare
- Extensible to different architecture and systems like Intel IA-32 and FreeBSD

fd.io (Fast Data Input/Output)

- Run by LFN - The LF(Linux Foundation) Networking Fund
- Cisco has donated VPP(Vector Packet Processing) library to fd.io
  - This library has been in production by Cisco since 2003
- Leverages DPDK capabilities
- Aligned to support NFV and SDN
- OPNFV is a sub-project of fd.io

netmap

- A novel framework which utilizes known techniques to reduce packet-processing costs
- A fast packet I/O mechanism between the NIC and user-space
  - Removes unnecessary metadata (e.g. sk_buf) allocation
  - Amortized systemcall costs, reduced/removed data copies
- Supported both in FreeBSD and Linux as loadable kernel module
- Comes as default from FreeBSD 11.0
- Released with BSD-2CLAUSE; FreeBSD is the primary development platform
- Supported with Intel, Realtek and Chelsio cards
- 14.8 Mpps achieved in 10G NIC with a 900mhz CPU
- Chelsio has tested 100G traffic in netmap mode with 99.99% success rate


Tx Rate (Mpps)
16
14
12
10
8
6
4
2
0
0
0.5
1
1.5
2
2.5
3
Clock speed (GHz)
netmap on 4 cores
netmap on 2 cores
netmap on 1 core
Linux/pktgen
FreeBSD/netsend

Other ecosystems

- OpenOnload by Solarflare
- Napatech

References

- pf_ring https://www.ntop.org
- DPDK https://www.dpdk.org
- fd.io https://fd.io
- netmap http://info.iet.unipi.it/~luigi/netmap/

Questions
Thank You