Beyond POSIX

Adventures in alternative networking APIs

Agenda

- Aeron Overview
- BSD Sockets
- Linux
    - recvmmsg/sendmmsg
    - io_uring
    - AF_PACKET
    - AF_XDP
- DPDK
- Others...

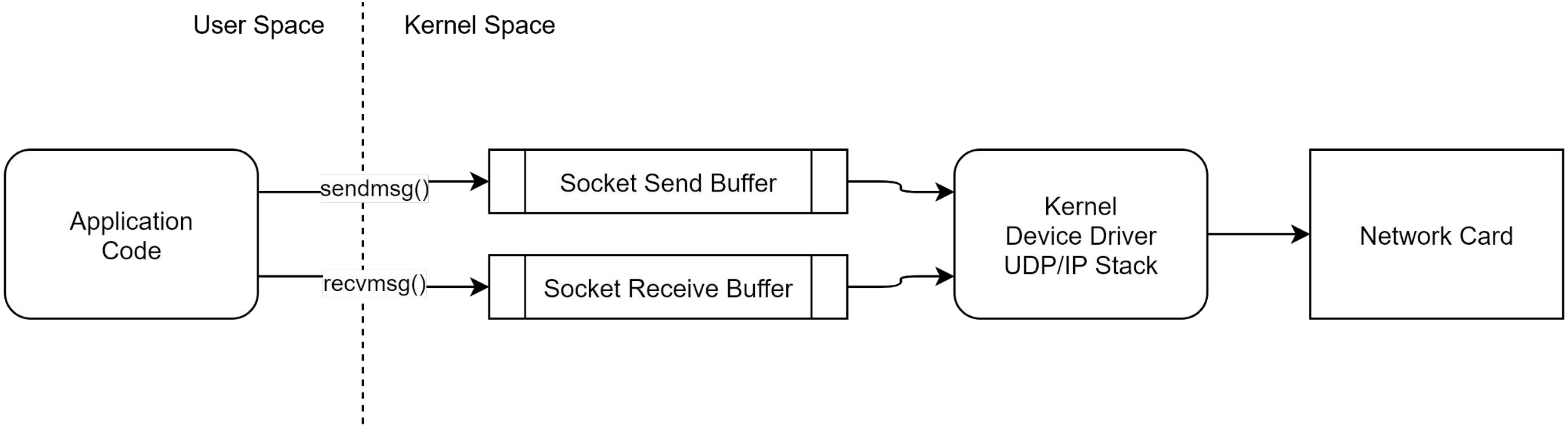
Aeron

- Layer 4 transport
- Message based
- Connection oriented
- Reliable
- Multicast
- Flow control
- Congestion control

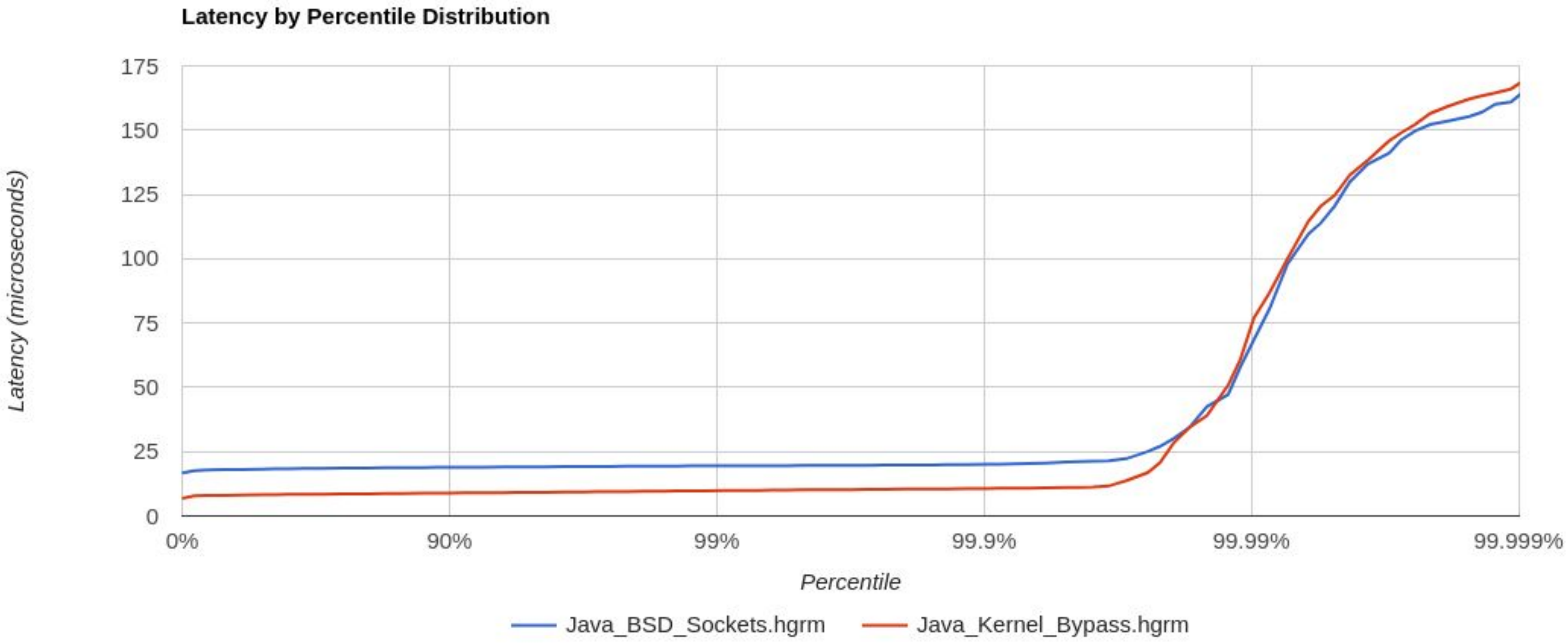
BSD Sockets

```
int socket_fd = socket(AF_INET, SOCK_DGRAM, 0);

struct sockaddr local_address;
bind(socket_fd, &local_address, sizeof(local_address));

int reuse = 1;
setsockopt(socket_fd, SOL_SOCKET, SO_REUSEADDR, &reuse, sizeof(reuse));

struct msghdr message_to_send;
sendmsg(socket_fd, &message_to_send, 0);

struct msghdr message_to_recv;
recvmsg(socket_fd, &message_to_recv, 0);
```

BSD Sockets


User Space
Kernel Space
Application Code
sendmsg()
recvmsg()
Socket Send Buffer
Socket Receive Buffer
Kernel
Device Driver
UDP/IP Stack
Network Card

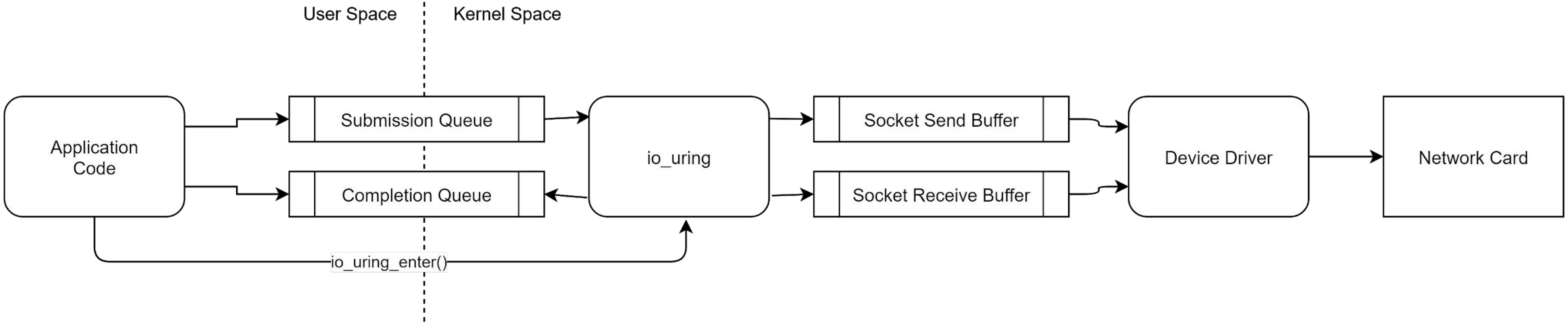
BSD Sockets


Latency by Percentile Distribution
175
150
125
100
75
50
25
0
Latency (microseconds)
0%
90%
99%
99.9%
99.99%
99.999%
Percentile
Java_BSD_Sockets.hgrm
Java_Kernel_Bypass.hgrm

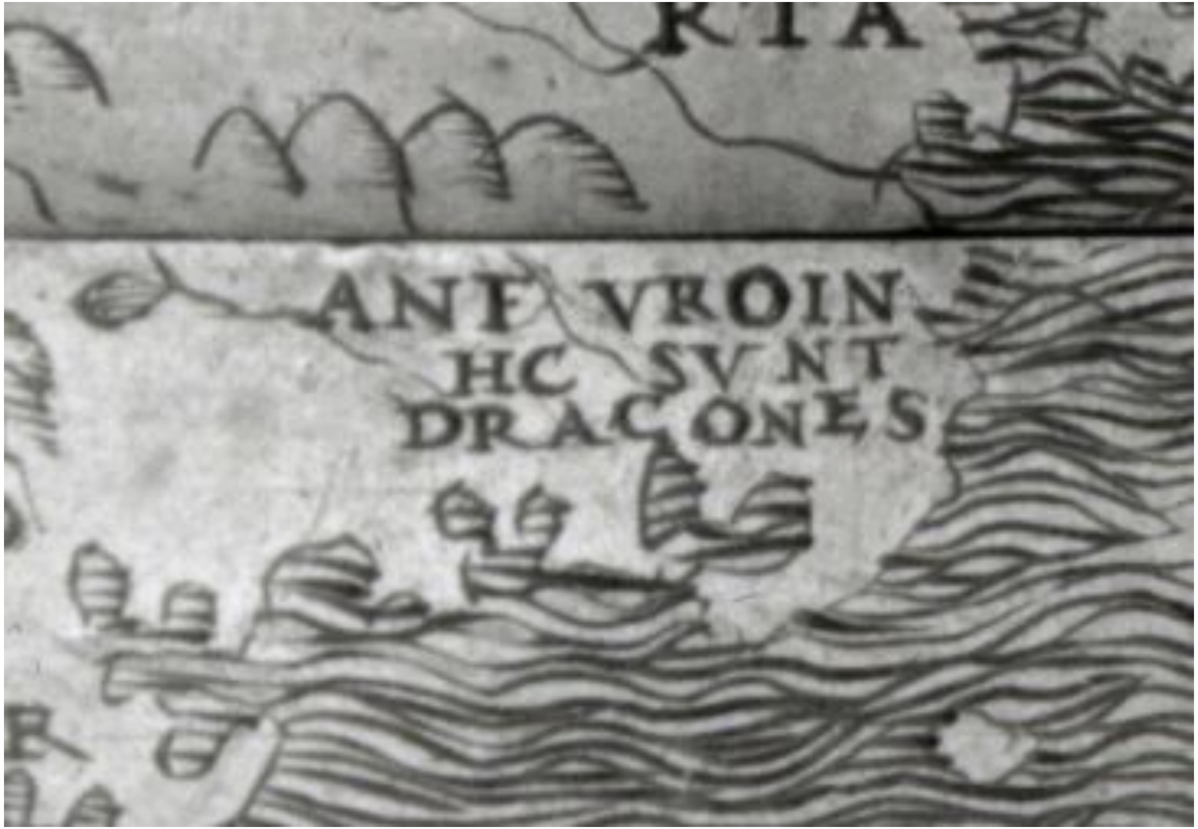
Linux recvmmsg/sendmmsg

```
struct sockaddr_in* address_1;
struct sockaddr_in* address_2;

struct mmsghdr messages_to_send[MAX_MESSAGES];
messages_to_send[0].msg_hdr.msg_name = address_1;
messages_to_send[1].msg_hdr.msg_name = address_2;

sendmmsg(socket_fd, message_to_send, 2);


struct mmsghdr messages_to_recv[MAX_MESSAGES];

recvmmsg(socket_fd, message_to_recv, MAX_MESSAGES);
```

Linux - io_uring



Here be Dragons!



UDP Header (Layer 4)

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
|           Src Port            |           Dest Port           |
+-------------------------------+-------------------------------+
|            Length             |          Checksum             |
+-------------------------------+-------------------------------+
|                             Data                            ...
...                                                             |
+--------------------------------------------------------------+
```

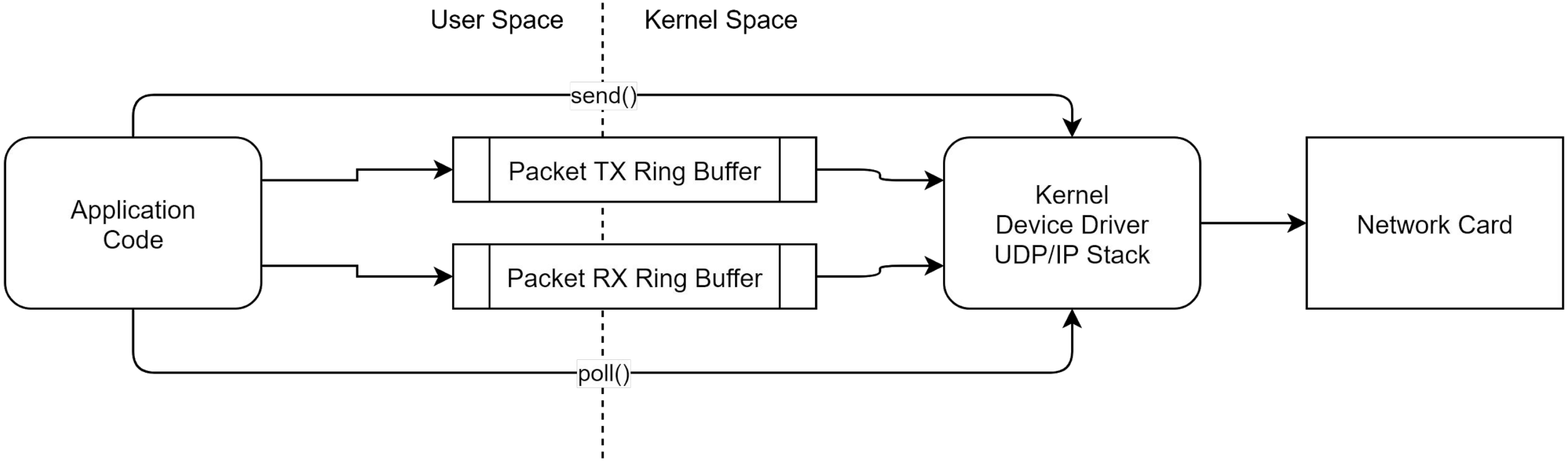
IP Header (Layer 3)

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------+-------+-----------+---+-------------------------------+
|Version|  IHL  | DSCP      |ECN|         Total Length          |
+-------+-------+-----------+---+-----+-------------------------+
|        Identification         |Flags|    Fragment Offset      |
+---------------+---------------+-----+-------------------------+
| Time to live  |    Protocol   |        Header Checksum        |
+---------------+---------------+-------------------------------+
|                         Src IP Address                        |
+---------------------------------------------------------------+
|                         Dest IP Address                       |
+---------------------------------------------------------------+
```

Ethernet Header (Layer 2)

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Dest MAC |  Src Mac  | T |
+-----------+-----------+---+
```

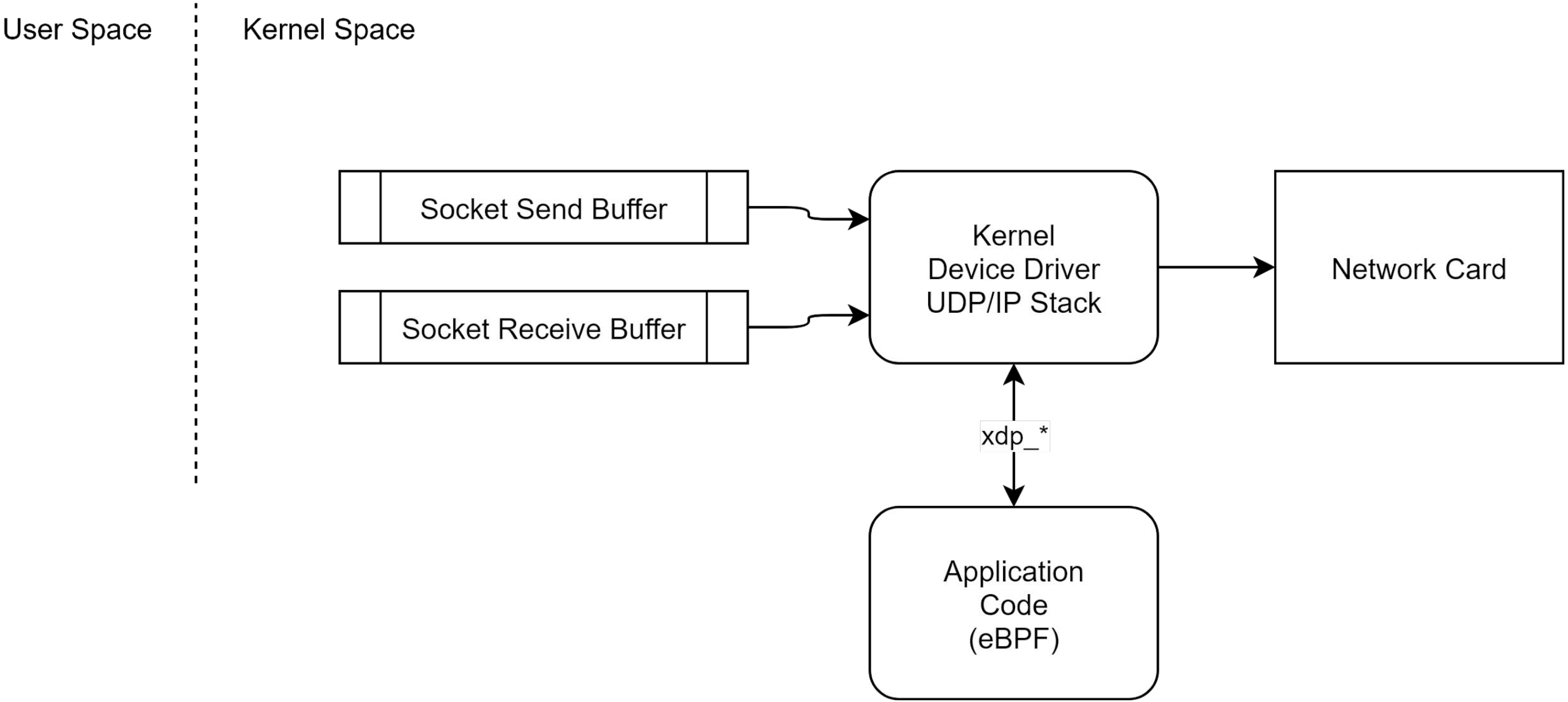
Linux - AF_PACKET / PACKET_MMAP



Linux - AF_PACKET / PACKET_MMAP

```
int v = TPACKET_V3
struct tpacket_req3 req;
// Some setup...

socket_fd = socket(AF_PACKET, SOCK_RAW, htons(ETH_P_ALL));

setsockopt(socket_fd, SOL_PACKET, PACKET_VERSION, &v, sizeof(v));
setsockopt(socket_fd, SOL_PACKET, PACKET_RX_RING, &req, sizeof(req));

void *map = mmap(
    NULL,
    req.tp_block_size * req.tp_block_nr,
    PROT_READ | PROT_WRITE,
    MAP_SHARED | MAP_LOCKED,
    socket_fd,
    0);
```

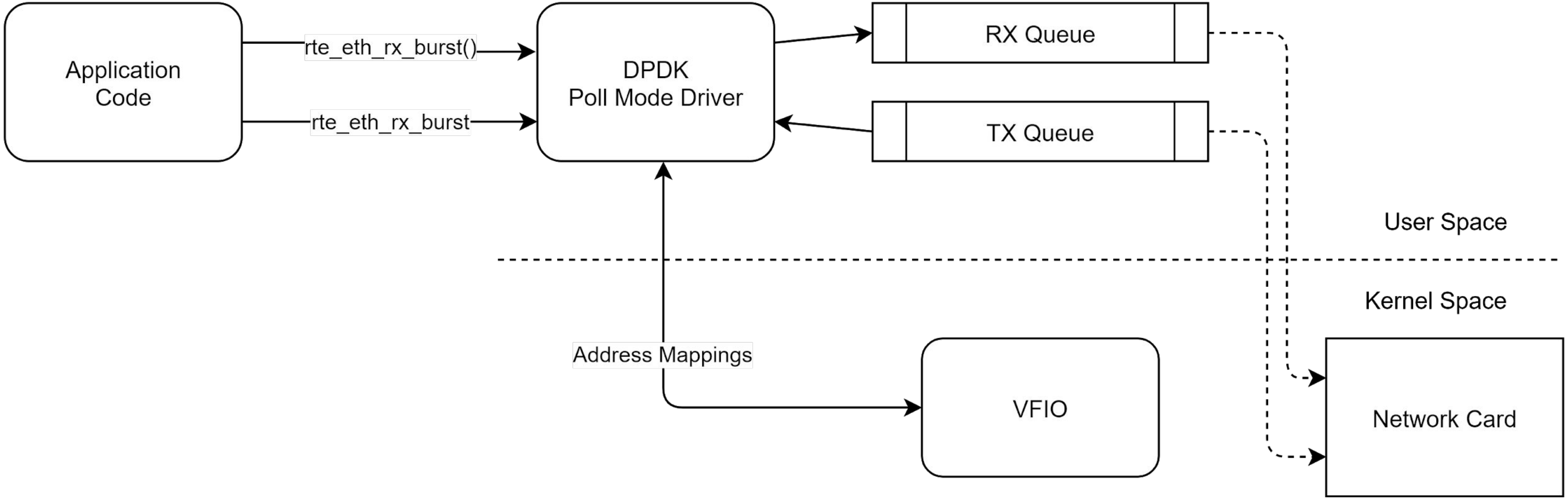
Linux - AF_XDP


User Space
Kernel Space
Socket Send Buffer
Socket Receive Buffer
Kernel
Device Driver
UDP/IP Stack
Network Card
xdp_*
Application
Code
(eBPF)

Linux - AF_XDP

```
SEC("prog")
int xdp_drop(struct xdp_md *ctx)
{
    //...
    char temp_ether_address[6];
    u32 temp_ip_address;
    u16 temp_port;

    if (echo)
    {
        memcpy(&temp_ether_address[0], &eth->h_source, sizeof(temp_ether_address));
        temp_ip_address = iph->saddr;
        temp_port = udph->source;

        memcpy(&eth->h_source, &eth->h_dest, sizeof(temp_ether_address));
        iph->saddr = iph->daddr;
        udph->source = udph->dest;

        memcpy(&eth->h_dest, &temp_ether_address[0], sizeof(temp_ether_address));
        iph->daddr = temp_ip_address;
        udph->dest = temp_port;

        return XDP_TX;
    }

    return XDP_PASS;
}
```

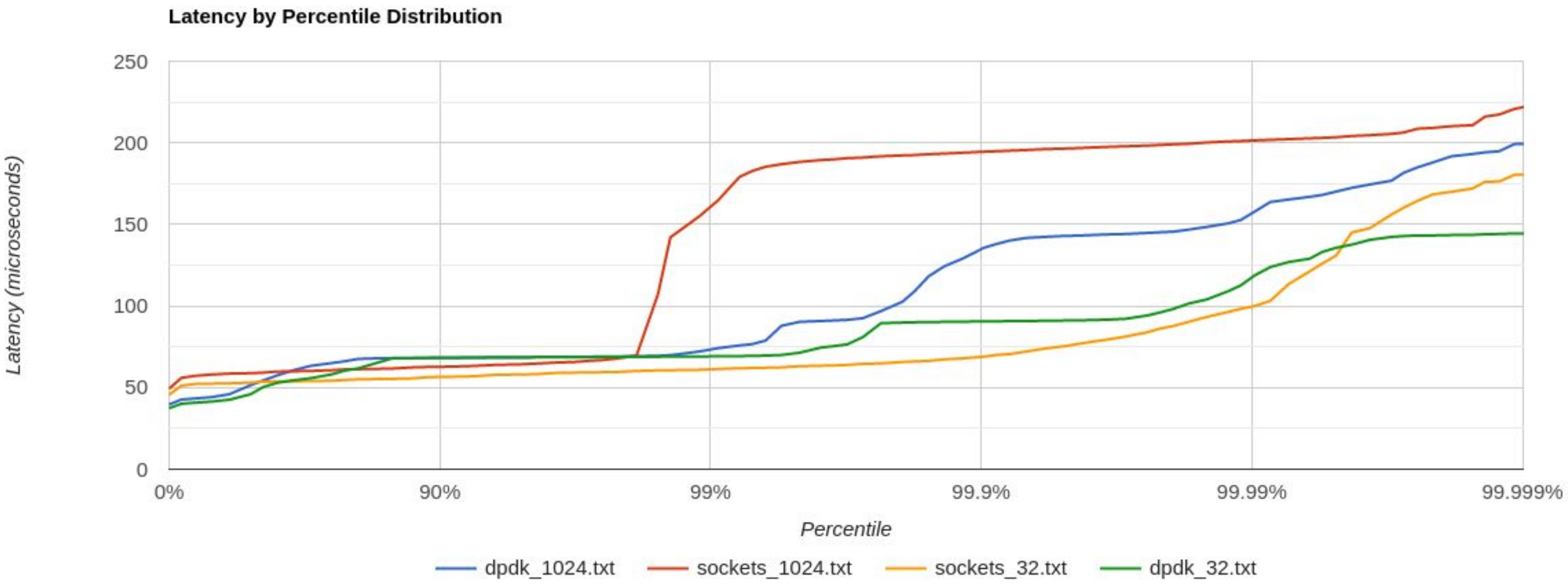
DPDK


Application Code
rte_eth_rx_burst()
rte_eth_rx_burst
DPDK Poll Mode Driver
RX Queue
TX Queue
User Space
Kernel Space
Address Mappings
VFIO
Network Card

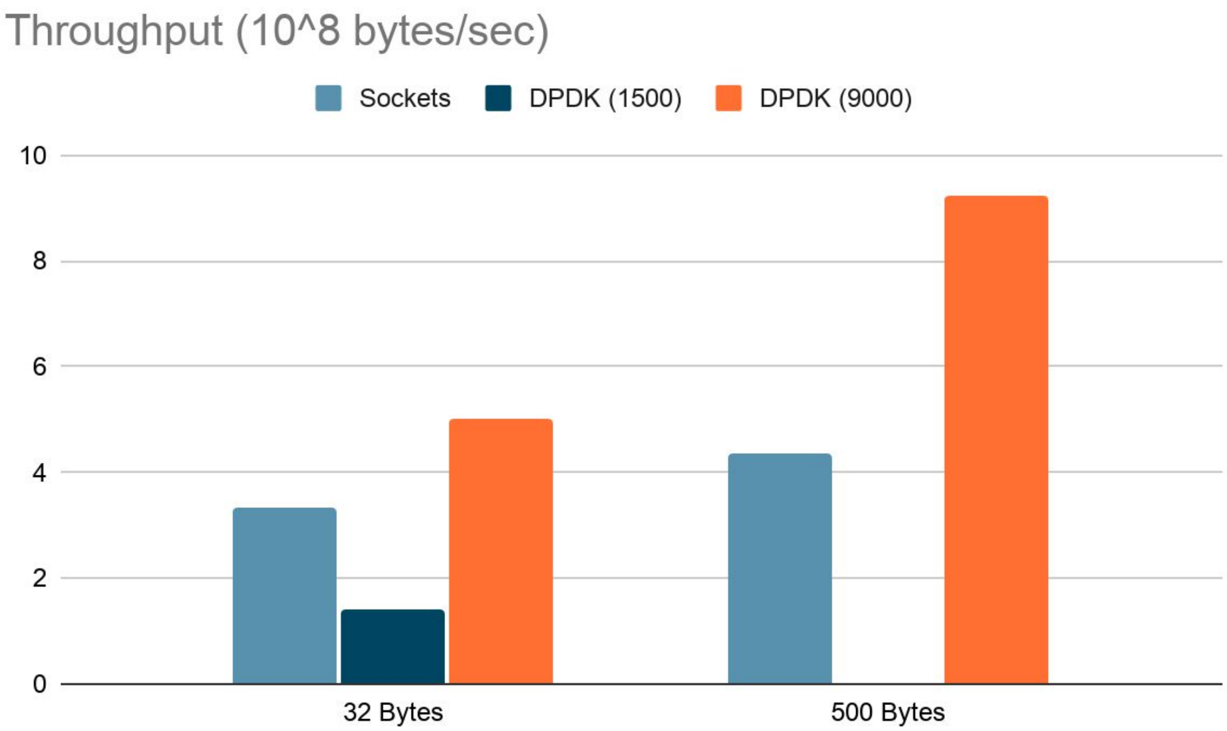
DPDK - Challenges

- IP Address -> HW Address mappings (ARP)
- Socket Buffers vs RX/TX queues
- Wildcard Ports / IN_ADDR_ANY
- Invisible to Kernel and associated tools, e.g. ip, ifconfig.
- MTU configuration
- Hardware offloading

DPDK Latency



DPDK Throughput



Others…

References

- sendmmsg: https://www.man7.org/linux/man-pages/man2/sendmmsg.2.html
- recvmmsg: https://www.man7.org/linux/man-pages/man2/recvmmsg.2.html
- io_uring: https://unixism.net/loti/what_is_io_uring.html
- Packet MMAP: https://www.kernel.org/doc/html/latest/networking/packet_mmap.html
- AF_XDP: https://www.kernel.org/doc/html/latest/networking/af_xdp.html
- DPDK: https://www.dpdk.org/
- OpenOnload: https://github.com/Xilinx-CNS/onload
- InfiniBand Verbs: https://www.mellanox.com/related-docs/prod_software/RDMA_Aware_Programming_user_manual.pdf
- Windows Registered I/O: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh997032(v=ws.11)