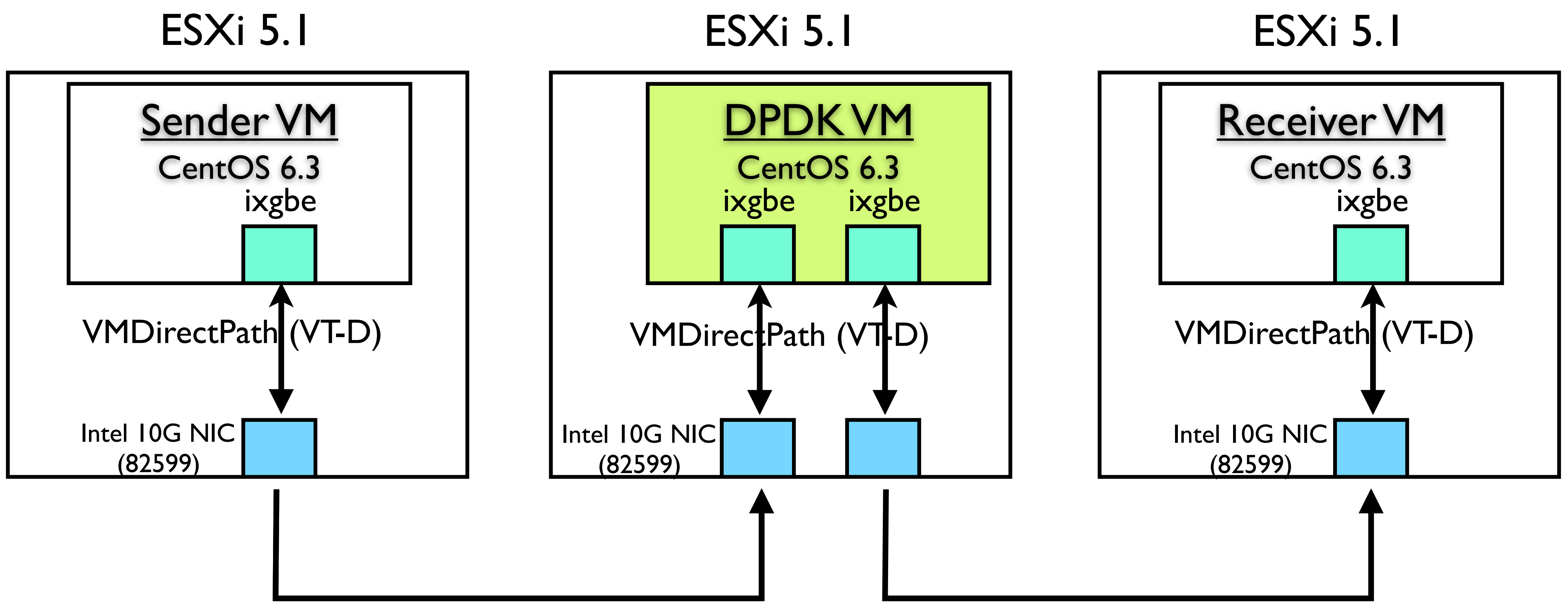
Intel DPDK
Step by Step Instructions

Hisaki Ohara (@hisak)

Objectives

- Build/Execute sample applications (helloworld, L2fwd and L3fwd)
- Packet forwarding by generating with Linux/pktgen

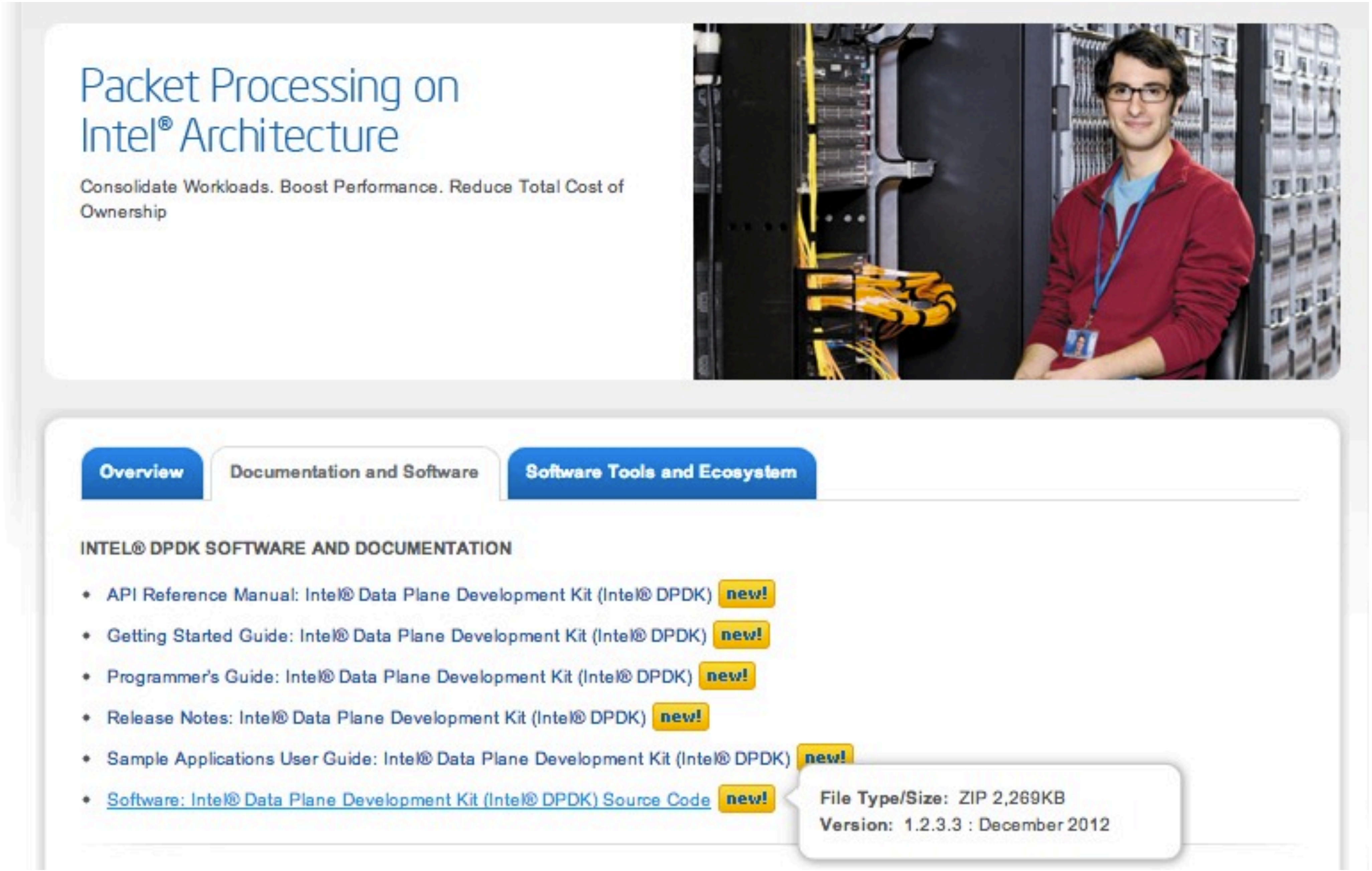
Test Environment



- ESXi 5.1
  - CPU: Xeon 5600 Series
  - Guest OS: CentOS 6.3 x86_64
    - # of vCPUs: 2
  - 10G NIC (82599) is passed through
    - In-box driver of ixgbe

Step0: Download source codes

- Source codes and relevant documents
  - http://www.intel.com/go/dpdk



Packet Processing on Intel® Architecture

Consolidate Workloads. Boost Performance. Reduce Total Cost of Ownership

INTEL® DPDK SOFTWARE AND DOCUMENTATION

- API Reference Manual: Intel® Data Plane Development Kit (Intel® DPDK) new!
- Getting Started Guide: Intel® Data Plane Development Kit (Intel® DPDK) new!
- Programmer's Guide: Intel® Data Plane Development Kit (Intel® DPDK) new!
- Release Notes: Intel® Data Plane Development Kit (Intel® DPDK) new!
- Sample Applications User Guide: Intel® Data Plane Development Kit (Intel® DPDK) new!
- Software: Intel® Data Plane Development Kit (Intel® DPDK) Source Code new!

File Type/Size: ZIP 2,269KB
Version: 1.2.3.3 : December 2012

Step1: Prepare Linux Kernel

- Add boot option and fstab for hugepage

```
# uname -a
Linux cent-dpdk 2.6.32-279.14.1.el6.x86_64 #1 SMP Tue Nov 6 23:43:09 UTC 2012 x86_64
x86_64 x86_64 GNU/Linux
# cat /boot/grub/grub.conf
<snip>
title CentOS (2.6.32-279.14.1.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-279.14.1.el6.x86_64 ro root=/dev/mapper/vg_cent6-
lv_root rd_LVM_LV=vg_cent6/lv_swap rd_NO_LUKS LANG=en_US.UTF-8 rd_NO_MD
rd_LVM_LV=vg_cent6/lv_root SYSFONT=latarcyrheb-sun16  KEYBOARDTYPE=pc KEYTABLE=us
rd_NO_DM rhgb quiet crashkernel=auto hugepages=256
        initrd /initramfs-2.6.32-279.14.1.el6.x86_64.img
<snip>
# mkdir /hugepages
# cat /etc/fstab
<snip>
hugetlbfs              /hugepages              hugetlbfs rw,mode=0777  0 0
# reboot
```

- Confirm hugepage is enabled

```
# cat /proc/meminfo
<snip>
HugePages_Total:     256
HugePages_Free:      256
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
<snip>
```

Step2: Build DPDK and samples

```
$ unzip INTELDPDK.L.1.2.3_3.zip
$ cd DPDK
$ make install T=x86_64-default-linuxapp-gcc
$ pwd
/home/dpdktest/DPDK
$ cd examples/helloworld
$ RTE_SDK=/home/dpdktest/DPDK make
  CC main.o
  LD helloworld
  INSTALL-APP helloworld
  INSTALL-MAP helloworld.map
```

Step3: helloworld sample

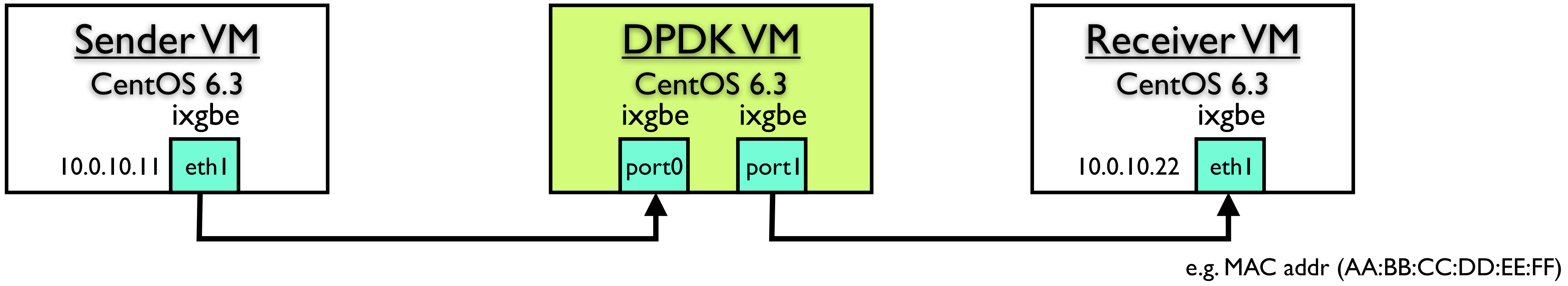
- Load required module for DPDK Linux app

```
# modprobe uio
# insmod /home/dpdktest/DPDK/x86_64-default-linuxapp-gcc/kmod/igb_uio.ko
```

- Execute helloworld sample

```
# ./build/helloworld -c 3 -n 2
EAL: coremask set to 3
EAL: Detected lcore 0 on socket 0
EAL: Detected lcore 1 on socket 0
EAL: Requesting 256 pages of size 2097152
EAL: Ask a virtual area of 0x20000000 bytes
EAL: Virtual area found at 0x7f4862c00000 (size = 0x20000000)
EAL: WARNING: Cannot mmap /dev/hpet! The TSC will be used instead.
EAL: Master core 0 is ready (tid=82dd0800)
EAL: Core 1 is ready (tid=621fe700)
hello from core 1
hello from core 0
```

Step4-1: L2fwd Sample Build



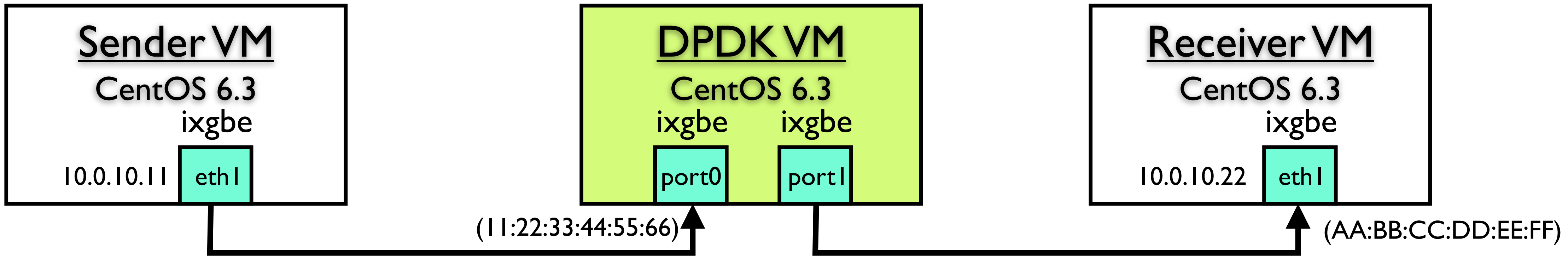
- L2fwd/L3fwd samples are very simple
  - One-way only. Don't expect ping/pong
  - Dest MAC address is hard-coded...

```
$ cd examples/l2fwd
$ diff -up main.c.0 main.c
@@ -293,7 +293,7 @@ l2fwd_simple_forward(struct rte_mbuf *m,

    /* 00:09:c0:00:00:xx */
        tmp = &eth->d_addr.addr_bytes[0];
-   *((uint64_t *)tmp) = 0x000000c00900 + (dst_port << 24);
+   *((uint64_t *)tmp) = 0xFFEEDDCCBBAA; /* AA:BB:CC:DD:EE:FF */

    /* src addr */
    ether_addr_copy(&l2fwd_ports_eth_addr[dst_port], &eth->s_addr);
$ RTE_SDK=/home/dpdktest/DPDK make
```

Step4-2: L2fwd Sample

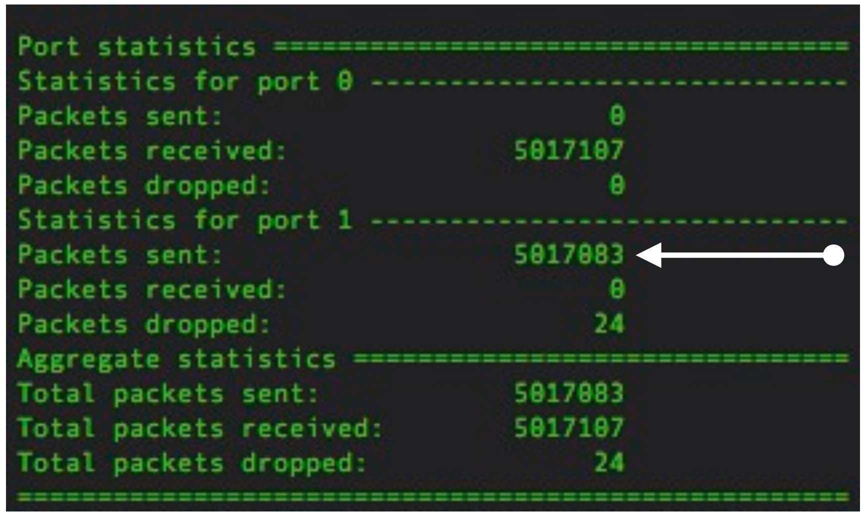


```
#ip address add 10.0.10.11/24 dev eth1

#modprobe pktgen
```

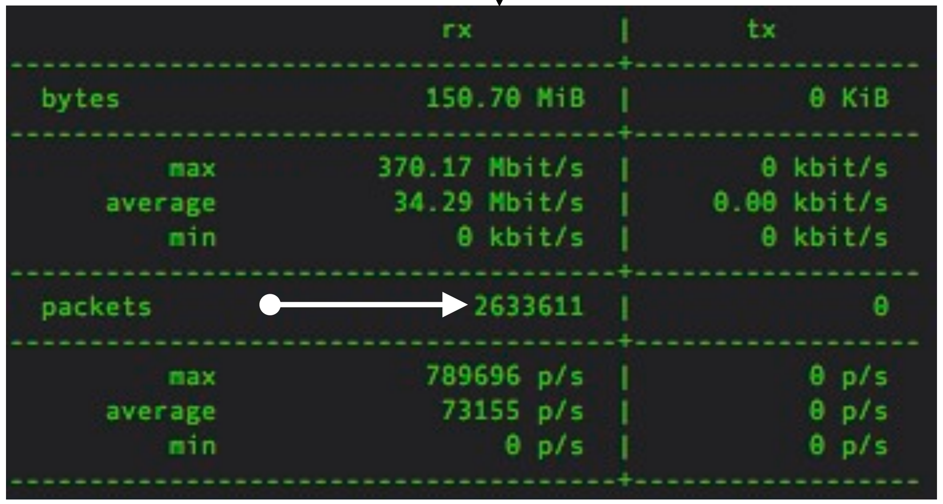
```
#echo “rem_device_all” > /proc/net/pktgen/kpktgend_0
#echo "add_device eth1" > /proc/net/pktgen/kpktgend_0
#echo "count 10000000" > /proc/net/pktgen/eth1
#echo "clone_skb 1000000" > /proc/net/pktgen/eth1
#echo "pkt_size 60" > /proc/net/pktgen/eth1
#echo "delay 0" > /proc/net/pktgen/eth1
#echo "dst 10.0.10.22" > /proc/net/pktgen/eth1
#echo "dst_mac 11:22:33:44:55:66" > /proc/net/pktgen/eth1
#echo "start" > /proc/net/pktgen/pgctrl
```

```
#./build/l2fwd -c 0x3 -n 2 -- -p 0x3
```



```
Port statistics ====================================
Statistics for port 0 ------------------------------
Packets sent:                          0
Packets received:                5017107
Packets dropped:                       0
Statistics for port 1 ------------------------------
Packets sent:                    5017083
Packets received:                      0
Packets dropped:                      24
Aggregate statistics ===============================
Total packets sent:              5017083
Total packets received:          5017107
Total packets dropped:                24
====================================================
```

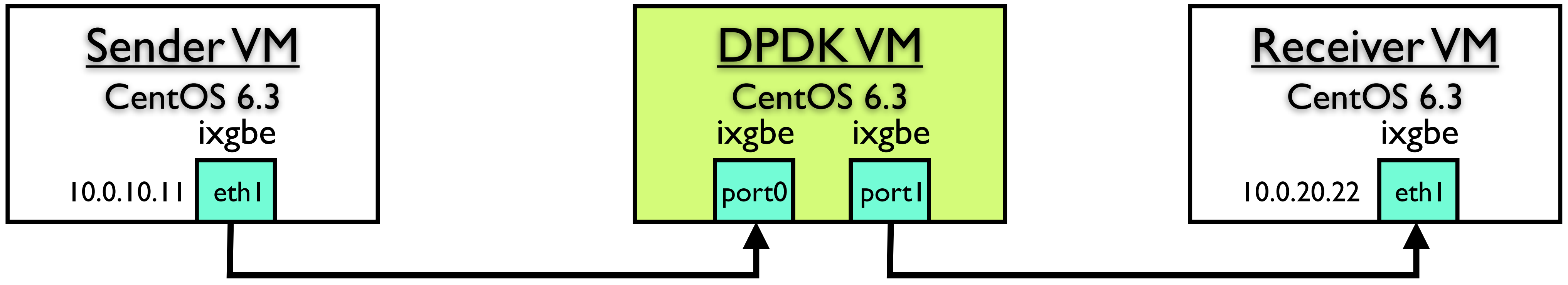
```
#ip address add 10.0.10.22/24 dev eth1
#vnstat -i eth1 -l
```



```
                              rx      |           tx
--------------------------------------+------------------
  bytes                   150.70 MiB  |          0 KiB
--------------------------------------+------------------
          max          370.17 Mbit/s  |     0 kbit/s
      average           34.29 Mbit/s  |  0.00 kbit/s
          min               0 kbit/s  |     0 kbit/s
--------------------------------------+------------------
  packets                    2633611  |              0
--------------------------------------+------------------
          max             789696 p/s  |          0 p/s
      average              73155 p/s  |          0 p/s
          min                  0 p/s  |          0 p/s
--------------------------------------+------------------
```

Packets are dropped at RX ports of DPDK VM and Receiver VM

Step5-1: L3fwd Sample Build



e.g. MAC addr (AA:BB:CC:DD:EE:FF)

- L3 fwd sample has two functions to determine destination port
  - [default] destination IP address (LPM-based)
  - 5-tuples (Hash-based)

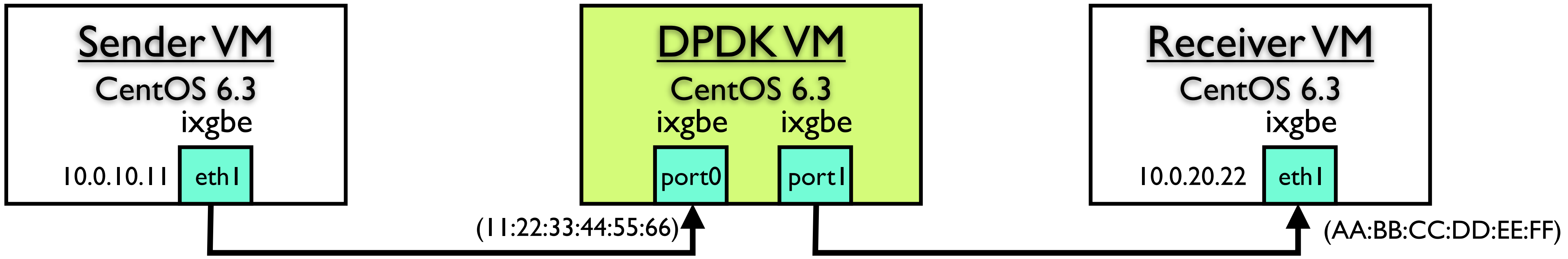
```
$ cd examples/l3fwd
$ diff -up main.c.0 main.c
@@ -282,6 +282,8 @@ static struct l3fwd_route l3fwd_route_ar
        {IPv4(6,1,1,0), 24, 5},
        {IPv4(7,1,1,0), 24, 6},
        {IPv4(8,1,1,0), 24, 7},
+       {IPv4(10,0,10,11), 24, 0},
+       {IPv4(10,0,20,22), 24, 1},
 };

 #define L3FWD_NUM_ROUTES \
@@ -475,7 +477,7 @@ l3fwd_simple_forward(struct rte_mbuf *m,

        /* 00:09:c0:00:00:xx */
        tmp = &eth_hdr->d_addr.addr_bytes[0];
-       *((uint64_t *)tmp) = 0x000000c00900 + (dst_port << 24);
+       *((uint64_t *)tmp) = 0xFFEEDDCCBBAA; /* AA:BB:CC:DD:EE:FF */

$ RTE_SDK=/home/dpdktest/DPDK make
```

Step5-2: L3fwd Sample



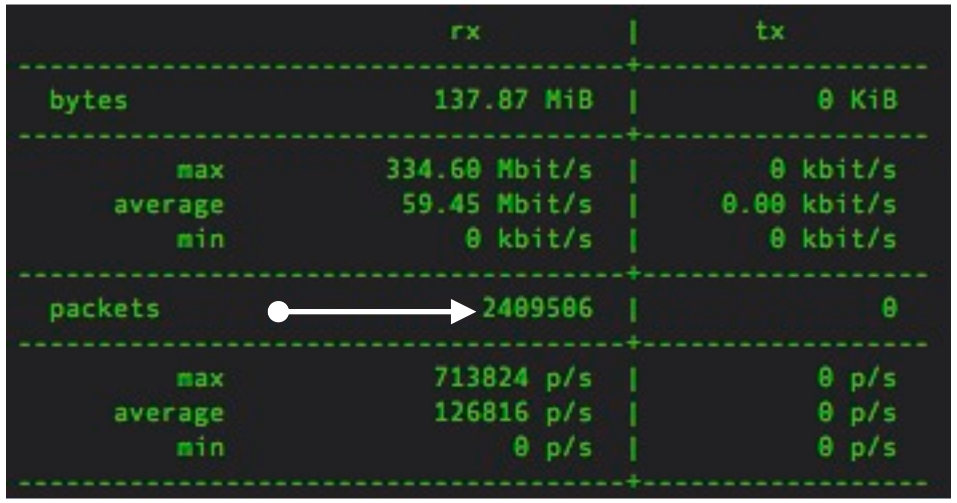
```
#ip address add 10.0.10.11/24 dev eth1

#modprobe pktgen
```

```
#./build/l3fwd -c 0x3 -n 2 -- -p 0x3
--config="(0,0,0),(1,0,1)"
```

```
#ip address add 10.0.20.22/24 dev eth1
#vnstat -i eth1 -l
```

```
#echo "rem_device_all" > /proc/net/pktgen/kpktgend_0
#echo "add_device eth1" > /proc/net/pktgen/kpktgend_0
#echo "count 10000000" > /proc/net/pktgen/eth1
#echo "clone_skb 1000000" > /proc/net/pktgen/eth1
#echo "pkt_size 60" > /proc/net/pktgen/eth1
#echo "delay 0" > /proc/net/pktgen/eth1
#echo "dst 10.0.20.22" > /proc/net/pktgen/eth1
#echo "dst_mac 11:22:33:44:55:66" > /proc/net/pktgen/eth1
#echo "start" > /proc/net/pktgen/pgctrl
```



```
                           rx         |        tx
--------------------------------------+------------------
  bytes                  137.87 MiB   |          0 KiB
--------------------------------------+------------------
          max          334.60 Mbit/s  |       0 kbit/s
      average           59.45 Mbit/s  |    0.00 kbit/s
          min               0 kbit/s  |       0 kbit/s
--------------------------------------+------------------
  packets                   2409506   |              0
--------------------------------------+------------------
          max              713824 p/s |            0 p/s
      average              126816 p/s |            0 p/s
          min                   0 p/s |            0 p/s
--------------------------------------+------------------
```

Need reliable ways and tunings
to measure performance

Notes on this experiment

- No guarantee as usual
- No tuning effort has been made
- References:
  - http://www.intel.com/go/dpdk
  - For pktgen (in Japanese)
    - http://research.sakura.ad.jp/2010/10/08/infini01/