Software Fastpath Service Quality Metrics Demo

# Introduction:

The **Software Fastpath Service Quality Metrics** project is focused on the development of utilities and libraries to support:

* Measurement of Telco Traffic and Performance KPIs through the low latency, high performance packet processing path (fast path) in the NFVI.
* Detection and reporting violations that can be consumed by VNFs and higher level EMS/OSS systems.

This Software Fastpath **Service Quality** Metrics Demo (SFQMD) zones in on the measurement of Telco Traffic and Performance KPIs in the fastpath and in particular the exposure of NIC MAC/PHY Level error and discard statistics counters of Niantic.

# Overview:



# HW & SW requirements:

|  |
| --- |
| **Landing Server** |
| IP address | 10.4.1.1 |
| Username | user |
| Password | user |
| SSH connect |

|  |
| --- |
| **SUT (Fedora 20 Kernel: 3.18.7-100.fc20.x86\_64 )** |
| IP address | 10.4.1.2 |
| Username | user |
| Password | user |
| SSH connect |

|  |
| --- |
| Ixia Chassis |
| IP address | 10.4.1.3 |
| Username | ixia |
| Password | P0222619 |
| Use RDP to connect |

|  |
| --- |
| **Ixia client VM** |
| IP address | 10.4.2.0 |
| Username | inteleil\administrator |
| Password | P@ssw0rd |
| Use RDP to connect |

|  |
| --- |
| **Software** |
| DPDKDPDK commit to apply demo patch to: | git clone <http://dpdk.org/git/dpdk>git checkout 2250cc5a191906c914221ff4f0da7b5d699b4175 |
| Demo patch for exposing the MAC&PHY registers |  |
| setup\_sfmd.sh: bash script for quick demo setup  | Please see Appendix A: |

# Log in:

Please see OPNFV Intel HF testbed – quickstart (VPN).docx

# SUT setup:

To Build DPDK you will need:

1. gcc:

“sudo yum install gcc”

1. General development tools:

“sudo yum groupinstall "Development Tools"

1. Matching kernel version packages:

“sudo yum install kernel kernel-devel kernel-headers kernel-modules-extra”

Once those kernel packages are installed you will need to update grub to boot into the new kernel at setup time:

Search for the kernel entry you just installed:

sudo grep ^menuentry /boot/grub2/grub.cfg | cut -d "'" -f2

Set the default kernel to the kernel you just installed:

sudo grub2-set-default 'Fedora (3.18.7-100.fc20.x86\_64) 20 (Heisenbug)'

List the kernel version to boot into by default:

sudo grub2-editenv list

For DPDK you will also need to modify grub to reserve some 1Gb hugpages on startup, to do this:

Edit the grub GRUB\_CMDLINE\_LINUX option to reserve some 1Gb hugepages:

sudo vi /etc/default/grub

|  |
| --- |
| **GRUB\_CMDLINE\_LINUX="**rd.lvm.lv=fedora/swap vconsole.font=latarcyrheb-sun16 rd.lvm.lv=fedora/root $([ -x /usr/sbin/rhcrashkernel-param ] && /usr/sbin/rhcrashkernel-param || :) rhgb quiet default\_hugepagesz=1G hugepagesz=1G hugepages=8" |

 Rebuild your grub config file:

 sudo grub2-mkconfig -o /boot/grub2/grub.cfg

 Append /etc/fstab to mount hugepages on startup

sudo vi /etc/fstab

|  |
| --- |
| nodev /mnt/huge\_1GB hugetlbfs pagesize=1GB 0 0 |

Now you are ready to reboot your machine into the new kernel.

# Demo setup on SUT:

Download DPDK:

git clone <http://dpdk.org/git/dpdk>

Checkout commit 2250cc5a191906c914221ff4f0da7b5d699b4175:

git checkout 2250cc5a191906c914221ff4f0da7b5d699b4175

Apply 0001-extending-ixgbe-rx-error-stats.patch:

Git apply 0001-extending-ixgbe-rx-error-stats.patch

Modify the setup\_sfmd.sh script to reflect the dpdk directory location and the Niantic ports PCI address, by default these values are set to:

RTE\_SDK=/home/user/dpdk

DPDK\_PORTS="04:00.0 04:00.1"

Setup the demo using setup\_sfqmd.sh (see Appendix A)

 ./setup\_sfmd.sh setup

This will build DPDK and bind the relevant Niantic ports to DPDK.

Run Testpmd:

 ./setup\_sfqmd.sh new

This will rebuild DPDK with flags to enable the DEMO for testpmd.

At the “testpmd” prompt, issue the following commands:

 testpmd > set fwd mac\_retry

 testpmd > start

testpmd > show port stats 0 OR testpmd > show port xstats 0

# IXIA Setup:

Add ports:



Add Chassis:



Select the ports and add them:



Setup 2 MAC interfaces on the ports by selecting the “static” configuration option under protocols, select the LANs tab and right click as shown in the figure below



Select the new option and add both ports:



Give each port a MAC address and enable both ports:



Add a quick test by selecting the QuickTests option as shown below:



Select the RFC 2889 Frame Error Filtering test with a new configuration



Both ports should already be selected. In the next tab so click next

Select the option to insert the IP Header with a valid checksum and change the destination IP address from the loopback address



Select your end points as shown below then click the green arrow:



Ensure that the values selected in the figure below match the values in your quick test:



Complete the setup by hitting next and finish on the next 2 windows.

Start the test:



Watch the error statistics counters increment on the SUT running testpmd:



# Appendix A (setup\_sfqmd.sh):

setup\_sfqmd.sh is a quick setup script for SFMD:

|  |
| --- |
| #!/bin/bashOLD="old"NEW="new"SETUP="setup"RTE\_SDK=/home/user/dpdkDPDK\_PORTS="04:00.0 04:00.1"if [ "$1" = $OLD ]; then cd $RTE\_SDK && make clean && make -j 23 install T=x86\_64-native-linuxapp-gccelif [ "$1" = $NEW ]; then cd $RTE\_SDK && make clean && make -j 23 install T=x86\_64-native-linuxapp-gcc EXTRA\_CFLAGS='-DDISPLAY'elif [ "$1" = $SETUP ]; then sudo ./tools/dpdk\_nic\_bind.py -b ixgbe $DPDK\_PORTS sudo rmmod igb\_uio && sudo rmmod uio cd $RTE\_SDK && make clean && make -j 23 install T=x86\_64-native-linuxapp-gcc sudo modprobe uio sudo insmod x86\_64-native-linuxapp-gcc/kmod/igb\_uio.ko sudo ./tools/dpdk\_nic\_bind.py -b igb\_uio $DPDK\_PORTS sudo ./tools/dpdk\_nic\_bind.py --status exitelse echo "Error Unrecognised Option!!\n" exitficd $RTE\_SDK/x86\_64-native-linuxapp-gcc/build/app/test-pmd && sudo ./testpmd -c 0x3 -n 4 -- --burst=64 -i |