# OPNFV Project Proposal – Telco KPI Monitoring

## Version: 1.0 (October 22nd, 2014)

## Project Name:

* Telco KPI Monitoring and Enforcement
* Proposed name for the repository: OPNFV\_TelcoKPIs
* Project Categories:
  + Documentation
  + Requirements
  + Integration & Testing
  + Collaborative Development

## Project description:

The OPNFV platform (NFVI) requires functionality to:

* Create a low latency, high performance packet processing path (fast path) through the NFVI that VNFs can take advantage of
* Measure Telco Traffic and Performance KPIs through that fast path
* Detect and report violations that can be consumed by VNFs and higher level EMS/OSS systems

This project is focused on the development of utilities and libraries to support the above. An ability to measure and enforce Traffic Quality KPI’s (Service Assurance) in the data-plane will be mandatory for any Telco-grade NFVI Implementation, regardless of whether the data plane is implemented in hardware or software.

Examples of local measurable QoS factors for Traffic Monitoring which impact both Quality of Experience and 5’9s availability would be (using Metro Ethernet Forum Guidelines as reference):

* Packet loss
* Packet Delay Variation
* Uni-directional frame delay

Other KPIs such as Call drops, Call Setup Success Rate, Call Setup time etc. are measured by the VNF.

In addition to Traffic Monitoring, the NFVI must also support Performance Monitoring of the physical interfaces themselves (e.g. NICs), i.e. an ability to monitor and trace errors on the physical interfaces and report them.

All these traffic statistics for Traffic and Performance Monitoring must be measured in-service and must be capable of being reported by standard Telco mechanisms (e.g. SNMP traps), for potential enforcement actions.

## Scope:

The output of the project will provide interfaces and functions to support monitoring of Packet Latency and Network Interfaces while the VNF is in service.

The DPDK interface/API will be updated to support:

1. Exposure of NIC MAC/PHY Level Counters
2. Interface for Time stamp on RX
3. Interface for Time stamp on TX

Specific testing and integration will be carried out to cover:

* Unit/Integration Test plans: A sample application provided to demonstrate packet latency monitoring and interface monitoring

The following list of features and functionality will be developed.

1. DPDK APIs and functions for latency and interface monitoring
2. A sample application to demonstrate usage

The scope of the project is limited to the DPDK APIs and a sample application to demonstrate usage.

In the diagram below, the interfaces 1, 2, 3 are implemented along with a sample application. The sample application will support monitoring of NIC counters/status and will also support measurement of packet latency using DPDK provided interfaces.

VNF specific processing, Traffic Monitoring, Performance Monitoring and Management Agent are out of scope. The scope is limited to Intel 10G Niantic support. The Proposed MAC/PHY Interface Counters include:

* Packet RX
* Packet TX
* Packet loss
* Interface errors + other stats

The Proposed Packet Latency Monitor include:

* Cycle accurate ‘stamping’ on ingress
* Supports latency measurements on egress

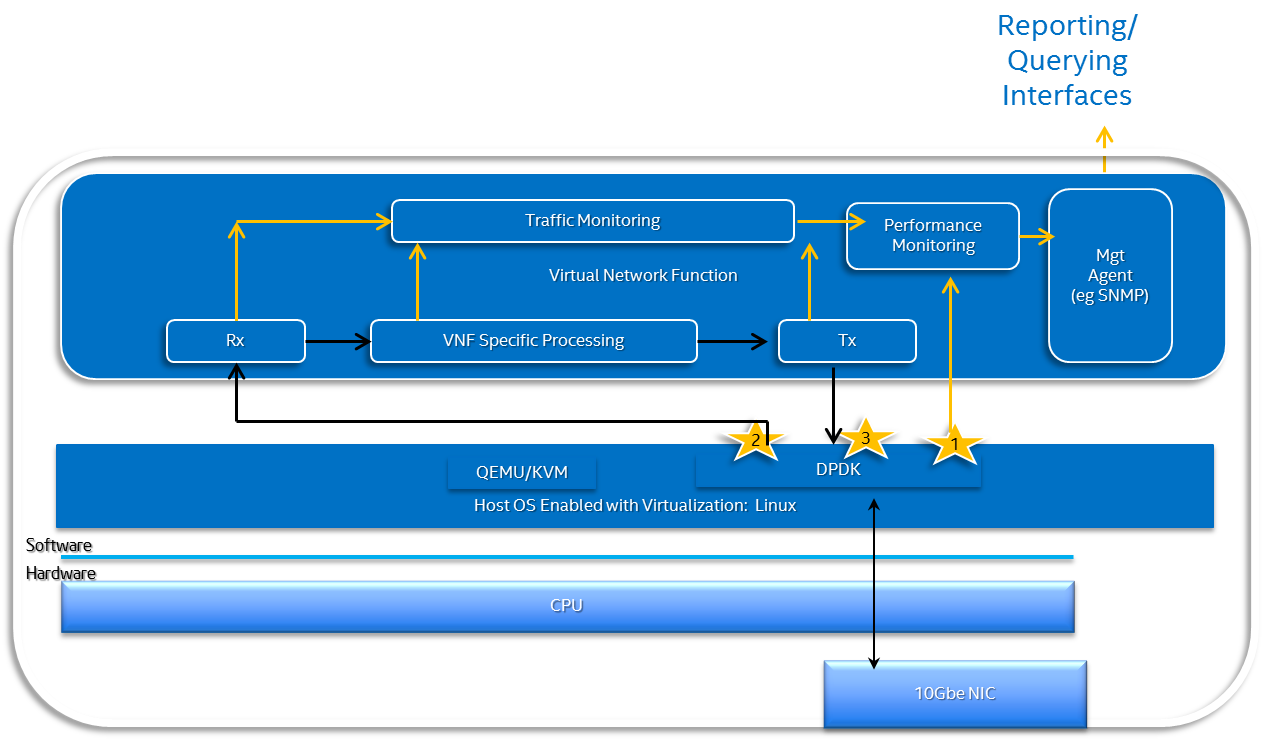


Fig 1: Architecture overview

Support for additional types of Network Interfaces can be added in the future.

### Testability: (optional, Project Categories: Integration & Testing)

Scalability: 2x10g Interface support

Tests: Traffic patterns include small packets (64 bytes) and average packet sizes of 200bytes.

There will be a specific testing resource assigned to the project

### Documentation:

* + Functional Architecture Specification
  + Release Notes
  + Programming Guide
  + API Reference Guide
  + Sample Application User Guide

## Dependencies:

* There are currently no similar projects underway or proposed in either OPNFV or in an upstream project.
* The relevant upstream project for this contribution is [www.dpdk.org](http://www.dpdk.org) and the aim is to intersect DPDK R2.0 with contribution patches including this functionality. DPDK R2.0 is provisionally planned for end March 2015.
* In terms of external fora or standard development organization dependencies to the project, there are no hard dependencies. The functionality developed by the project is designed to be consumed by existing northbound interfaces and implementations.
* For the integration and test requirements, there is a dependency towards IA server Hardware and NICs being available to the committers.

## Committers and Contributors:

* Maintainers: Thomas Monjalon, [www.dpdk.org](http://www.dpdk.org) maintainer; email: [Thomas.monjalon@6wind.com](mailto:Thomas.monjalon@6wind.com)
* Committers: Bruce Richardson: [Bruce.richardson@intel.com](mailto:Bruce.richardson@intel.com); Maryam Tahhan: [maryam.tahhan@intel.com](mailto:maryam.tahhan@intel.com)
* Other contributors: John Browne: [john.j.browne@intel.com](mailto:john.j.browne@intel.com); Mike Lynch: [Michael.a.lynch@intel.com](mailto:Michael.a.lynch@intel.com);

## Planned deliverables:

* The project release package will be an upstream contribution into [www.dpdk.org](http://www.dpdk.org)
* There will be the following updates to dpdk.org collateral to support the contribution:
  + Updates to DPDK R2.0 Release Notes
  + Updates to DPDK R2.0 Programming Guide
  + Updates to DPDK R2.0 API Reference Guide
  + Updates to DPDK R2.0 Sample Applications User Guide

## Proposed Release Schedule:

* The first release of this functionality is planned to align with DPDK 2.0 in March 2015, which should align with OPNFV R1.0 timeframe of 1H 2015.