



# VSPERF Guides and Installation

*Release 1.0.0*

**OPNFV**

November 26, 2015



<b>1</b>	<b>Getting Started with ‘vsperf’</b>	<b>1</b>
1.1	Hardware Requirements . . . . .	1
1.2	vSwitch Requirements . . . . .	1
1.3	Installation . . . . .	1
1.4	IXIA Setup . . . . .	1
1.5	Spirent Setup . . . . .	2
1.6	Cloning and building src dependencies . . . . .	3
1.7	Configure the <code>./conf/10_custom.conf</code> file . . . . .	4
1.8	Using a custom settings file . . . . .	4
1.9	Executing tests . . . . .	4
1.10	Executing Vanilla OVS tests . . . . .	5
1.11	Executing PVP tests using Vanilla OVS . . . . .	6
1.12	Code change verification by pylint . . . . .	6
1.13	GOTCHAs: . . . . .	7
<b>2</b>	<b>Installing vswitchperf</b>	<b>9</b>
<b>3</b>	<b>Working Behind a Proxy</b>	<b>11</b>



## GETTING STARTED WITH 'VSPERF'

### 1.1 Hardware Requirements

VSPERF requires one of the following traffic generators to run tests:

- IXIA traffic generator (IxNetwork hardware) and a machine that runs the IXIA client software
- Spirent traffic generator (TestCenter hardware chassis or TestCenter virtual in a VM) and a

VM to run the Spirent Virtual Deployment Service image, formerly known as "Spirent LabServer".

Both test configurations, above, also require a CentOS Linux release 7.1.1503 (Core) host.

### 1.2 vSwitch Requirements

The vSwitch must support Open Flow 1.3 or greater.

### 1.3 Installation

Follow the installation instructions to install.

### 1.4 IXIA Setup

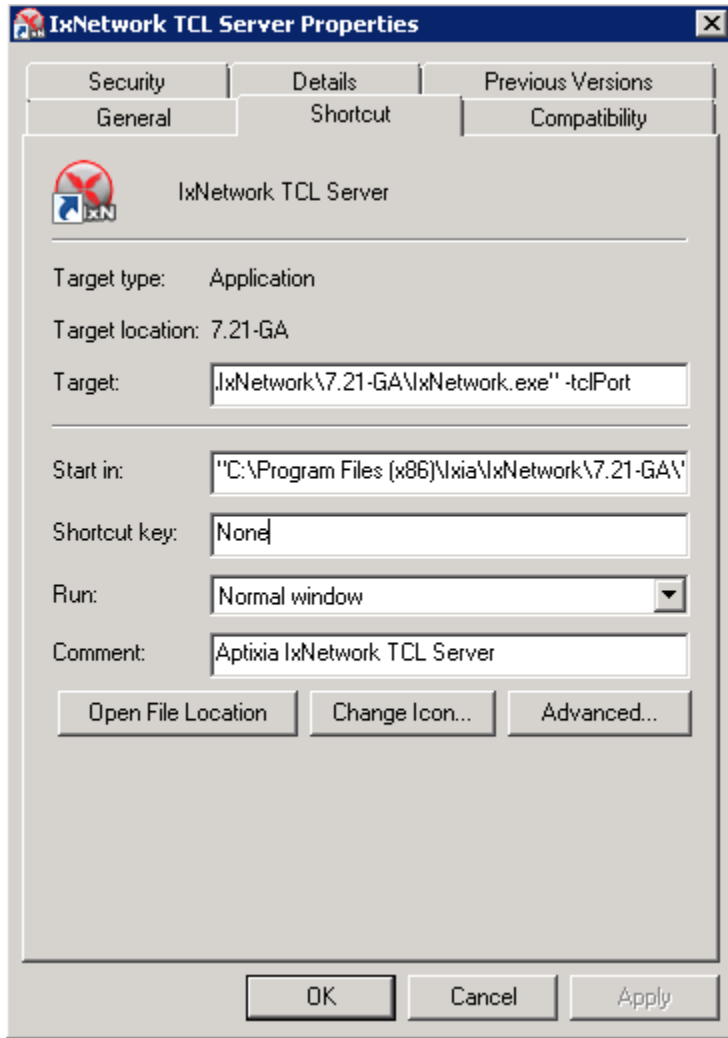
#### 1.4.1 On the CentOS 7 system

You need to install IxNetworkTclClient\$(VER\_NUM)Linux.bin.tgz.

#### 1.4.2 On the IXIA client software system

Find the IxNetwork TCL server app (start -> All Programs -> IXIA -> IxNetwork -> IxNetwork\_\$(VER\_NUM) -> IxNetwork TCL Server)

Right click on IxNetwork TCL Server, select properties - Under shortcut tab in the Target dialogue box make sure there is the argument "-tclport xxxx" where xxxx is your port number (take note of this port number you will need it for the 10\_custom.conf file).



Hit Ok and start the TCL server application

## 1.5 Spirent Setup

Spirent installation files and instructions are available on the Spirent support website at:

<http://support.spirent.com>

Select a version of Spirent TestCenter software to utilize. This example will use Spirent TestCenter v4.57 as an example. Substitute the appropriate version in place of 'v4.57' in the examples, below.

### 1.5.1 On the CentOS 7 System

Download and install the following:

Spirent TestCenter Application, v4.57 for 64-bit Linux Client

## 1.5.2 Spirent Virtual Deployment Service (VDS)

Spirent VDS is required for both TestCenter hardware and virtual chassis in the vsperf environment. For installation, select the version that matches the Spirent TestCenter Application version. For v4.57, the matching VDS version is 1.0.55. Download either the ova (VMware) or qcow2 (QEMU) image and create a VM with it. Initialize the VM according to Spirent installation instructions.

## 1.5.3 Using Spirent TestCenter Virtual (STCv)

STCv is available in both ova (VMware) and qcow2 (QEMU) formats. For VMware, download:

Spirent TestCenter Virtual Machine for VMware, v4.57 for Hypervisor - VMware ESX.ESXi

Virtual test port performance is affected by the hypervisor configuration. For best practice results in deploying STCv, the following is suggested:

- Create a single VM with two test ports rather than two VMs with one port each
- Set STCv in DPDK mode
- Give STCv  $2*n + 1$  cores, where  $n$  = the number of ports. For vsperf, cores = 5.
- Turning off hyperthreading and pinning these cores will improve performance
- Give STCv 2 GB of RAM

To get the highest performance and accuracy, Spirent TestCenter hardware is recommended. vsperf can run with either stype test ports.

## 1.6 Cloning and building src dependencies

In order to run VSPERF, you will need to download DPDK and OVS. You can do this manually and build them in a preferred location, or you could use vswitchperf/src. The vswitchperf/src directory contains makefiles that will allow you to clone and build the libraries that VSPERF depends on, such as DPDK and OVS. To clone and build simply:

```
cd src
make
```

VSPERF can be used with OVS without DPDK support. In this case you have to specify path to the kernel sources by WITH\_LINUX parameter:

```
cd src
make WITH_LINUX=/lib/modules/`uname -r`/build
```

To build DPDK and OVS for PVP and PVVP testing with vhost\_user as the guest access method, use:

```
make VHOST_USER=y
```

To build everything: Vanilla OVS, OVS with vhost\_user as the guest access method and OVS with vhost\_cuse access simply:

```
make
```

The vhost\_user build will reside in src/ovs/ The vhost\_cuse build will reside in vswitchperf/src\_cuse The Vanilla OVS build will reside in vswitchperf/src\_vanilla

To delete a src subdirectory and its contents to allow you to re-clone simply use:

```
make clobber
```

## 1.7 Configure the `./conf/10_custom.conf` file

The `10_custom.conf` file is the configuration file that overrides default configurations in all the other configuration files in `./conf`. The supplied `10_custom.conf` file must be modified, as it contains configuration items for which there are no reasonable default values.

The configuration items that can be added is not limited to the initial contents. Any configuration item mentioned in any `.conf` file in `./conf` directory can be added and that item will be overridden by the custom configuration value.

## 1.8 Using a custom settings file

If your `10_custom.conf` doesn't reside in the `./conf` directory or if you want to use an alternative configuration file, the file can be passed to `vsperf` via the `--conf-file` argument.

```
./vsperf --conf-file <path_to_settings_py> ...
```

Note that configuration passed in via the environment (`--load-env`) or via another command line argument will override both the default and your custom configuration files. This “priority hierarchy” can be described like so (1 = max priority):

1. Command line arguments
2. Environment variables
3. Configuration file(s)

---

## 1.9 Executing tests

Before running any tests make sure you have root permissions by adding the following line to `/etc/sudoers`:

```
username ALL=(ALL) NOPASSWD: ALL
```

`username` in the example above should be replaced with a real username.

To list the available tests:

```
./vsperf --list
```

To run a single test:

```
./vsperf $TESTNAME
```

Where `$TESTNAME` is the name of the `vsperf` test you would like to run.

To run a group of tests, for example all tests with a name containing ‘RFC2544’:

```
./vsperf --conf-file=user_settings.py --tests="RFC2544"
```

To run all tests:



```
./vsperf --conf-file=user_settings.py
```

Some tests allow for configurable parameters, including test duration (in seconds) as well as packet sizes (in bytes).

```
./vsperf --conf-file user_settings.py
--tests RFC2544Tput
--test-param "duration=10;pkt_sizes=128"
```

For all available options, check out the help dialog:

```
./vsperf --help
```

## 1.10 Executing Vanilla OVS tests

If you have compiled all the variants of OVS in ‘src/’ please skip step 1.

### 1. Recompile src for Vanilla OVS testing

```
cd src
make cleanse
make WITH_LINUX=/lib/modules/`uname -r`/build
```

### 2. Update your “10\_custom.conf” file to use the appropriate variables for Vanilla OVS:

```
VSWITCH = 'OvsVanilla'
VSWITCH_VANILLA_PHY_PORT_NAMES = ['$PORT1', '$PORT1']
```

Where \$PORT1 and \$PORT2 are the Linux interfaces you’d like to bind to the vswitch.

### 3. Run test:

```
./vsperf --conf-file <path_to_settings_py>
```

Please note if you don’t want to configure Vanilla OVS through the configuration file, you can pass it as a CLI argument; BUT you must set the ports.

```
./vsperf --vswitch OvsVanilla

Executing PVP and PVVP tests
```

### 1. Set VHOST\_METHOD and VNF of your settings file to:

```
VHOST_METHOD='user'
VNF = 'QemuDpdkVhost'
```

### 2. Recompile src for VHOST USER testing

```
cd src
make cleanse
make VHOST_USER=y
```

### 3. Run test:

```
./vsperf --conf-file <path_to_settings_py>
```

To run tests using vhost-cuse as guest access method:

### 1. Set VHOST\_METHOD and VNF of your settings file to:

```
VHOST_METHOD='cuse'  
VNF = 'QemuDpdkVhostCuse'
```

### 2. Recompile src for VHOST USER testing

```
cd src  
make cleanse  
make VHOST_USER=n
```

### 3. Run test:

```
./vsperf --conf-file <path_to_settings_py>
```

## 1.11 Executing PVP tests using Vanilla OVS

To run tests using Vanilla OVS:

### 1. Set the following variables:

```
VSWITCH = 'OvsVanilla'  
VNF = 'QemuVirtioNet'  
  
VANILLA_TGEN_PORT1_IP = n.n.n.n  
VANILLA_TGEN_PORT1_MAC = nn:nn:nn:nn:nn:nn  
  
VANILLA_TGEN_PORT2_IP = n.n.n.n  
VANILLA_TGEN_PORT2_MAC = nn:nn:nn:nn:nn:nn  
  
VANILLA_BRIDGE_IP = n.n.n.n  
  
or use --test-param  
  
./vsperf --conf-file user_settings.py  
--test-param "vanilla_tgen_tx_ip=n.n.n.n;  
vanilla_tgen_tx_mac=nn:nn:nn:nn:nn:nn"
```

### 2. Recompile src for Vanilla OVS testing

```
cd src  
make cleanse  
make WITH_LINUX=/lib/modules/`uname -r`/build
```

### 3. Run test:

```
./vsperf --conf-file <path_to_settings_py>
```

## 1.12 Code change verification by pylint

Every developer participating in VSPERF project should run pylint before his python code is submitted for review. Project specific configuration for pylint is available at 'pylint.rc'.

Example of manual pylint invocation:

```
pylint --rcfile ./pylintrc ./vsperf
```

## 1.13 GOTCHAs:

### 1.13.1 OVS with DPDK and QEMU

If you encounter the following error: “before (last 100 chars): ‘-path=/dev/hugepages,share=on: unable to map backing store for hugepages: Cannot allocate memoryrnrn” with the PVP or PVVP deployment scenario, check the amount of hugepages on your system:

```
cat /proc/meminfo | grep HugePages
```

By default the vswitchd is launched with 1Gb of memory, to change this, modify `--socket-mem` parameter in `conf/02_vswitch.conf` to allocate an appropriate amount of memory:

```
VSWITCHD_DPDK_ARGS = ['-c', '0x4', '-n', '4', '--socket-mem 1024,0']
```

---



## INSTALLING VSWITCHPERF

The test suite requires Python 3.3 and relies on a number of other packages. These need to be installed for the test suite to function. To install Python 3.3 in CentOS 7, an additional repository, Software Collections (see <https://www.softwarecollections.org/en/scls/rhscl/python33>) should be enabled.

Installation of required packages and preparation of Python 3 virtual environment is performed by `systems/build_base_machine.sh`. It should be executed under user account, which will be used for `vsperf` execution. Please Note: Password-less `sudo` access must be configured for given user account before script is executed.

Execution of installation script:

```
cd systems
./build_base_machine.sh
```

Please note: you don't need to go into any of the `systems` subdirectories, simply run the top level `build_base_machine.sh`, your OS will be detected automatically.

You will need to activate the virtual environment every time you start a new shell session. To activate, simple run:

```
scl enable python33 bash
cd $HOME/vsperfenv
source bin/activate
```



## WORKING BEHIND A PROXY

If you're behind a proxy, you'll likely want to configure this before running any of the above. For example:

```
export http_proxy=proxy.mycompany.com:123
export https_proxy=proxy.mycompany.com:123
```

Revision: 0606ef6290ac1e41468f57b0ef2ff1d0571aee3e

Build date: November 26, 2015