

# Traffic Generator Testing

- [Introduction](#)
- [Traffic Generator Categorization for VSPERF Tests](#)
- [Tests Run](#)
  - [Additional Test Configuration Information](#)
- [Analysis of Test Results](#)
- [Test Topology](#)
- [Test Results](#)
  - [Hardware Traffic Generator Results](#)
    - [RFC2544 OVS and VPP – Single Flow, Bidirectional](#)
    - [RFC2544 OVS and VPP Multi-Stream.](#)
  - [Baremetal \(BM-A\) Results](#)
    - [RFC2544 OVS and VPP – Single Flow, Bidirectional](#)
    - [RFC2544 OVS and VPP Multi-Stream.](#)
  - [Baremetal \(BM-B\) Results](#)
    - [RFC2544 OVS and VPP – Single Flow, Bidirectional](#)
    - [RFC2544 OVS and VPP Multi-Stream.](#)
  - [Traffic Generator as VM Results](#)
    - [RFC2544 OVS and VPP – Single Flow, Bidirectional](#)
    - [RFC2544 OVS and VPP Multi-Stream.](#)
  - [Noisy-Neighbor test results](#)
- [Future Works.](#)
- [Appendix-A: BM-B Re-Run](#)
- [Appendix B: Back2Back Testing Time Series \(from CI\)](#)
- [Appendix C: BM-C Traffigen Results](#)
- [Appendix D: Analysis of Cache-Misses](#)
- [Appendix E: Hardware and Baremetal Traffic Generators - Summarized](#)

## Introduction

VSPERF participated in OPNFV Plugfest for Danube release at Orange Gardens, Paris from 24<sup>th</sup> April 2017 to 28<sup>th</sup> April 2017. As part of the plugfest VSPERF had following activities planned:

1. Test activities
2. VSPERF integration activities
3. Discussion topics / planning activities

This page describes the (1) activity – Test execution – which can be categorized as follows:

- (a) NFVI Benchmarking - VPP vs OVS-dpdk: OVS vs VPP – Comparing the two virtual switches – in terms of both performance and resource-consumption
- (b) Traffic generator comparisons - Use VSPERF to compare test results with traffic generators. Section-2 explains more about the traffic generators used.
- (c) Stress tests - Noisy neighbor tests with a Stress-VM.

Summit Presentation:



## Traffic Generator Categorization for VSPERF Tests

<a href="#">Category</a>	<a href="#">A brief discription</a>
--------------------------	-------------------------------------

Hardware	Chassis from Hardware test-equipment vendor.
Baremetal	BM-A, BM-B and BM-C are packet generators built on top of DPDK.
VM	Traffic generator as a Virtual-Machine.

## Tests Run

Environment	Benchmarking Standard	VSPERF Test Code	Traffic Configuration	Traffic Generators	Results-Plot Reference
VSPERF with OVS as the vswitch	RFC2544, Throughput	phy2phy_tput	Bi-Directional Multistream=False	All.	Fig: 3, 4, 9, 10, 15, 17, 18.
VSPERF with VPP as the vswitch	RFC2544, Throughput	phy2phy_tput_vpp	Bi-Directional Multistream=False	All	
VSPERF with OVS as the vswitch	RFC2544, Throughput.	phy2phy_tput	Bi-Directional Multistream=True  Number of streams – 4096, 1M, 16.8M (BM-A Only)	All	Figs: 5, 6,7, 8, 11, 12, 13, 14, 16, 19, 20
VSPERF with VPP as the vswitch	RFC2544, Throughput	phy2phy_tput_vpp	Bi-Directional Multistream=True  Streams – 4096, 1M, 16.8M(BM-A Only)	All	
VSPERF with OVS as vSwitch and a VLoop VM.  One/Two Stress VMs.	RFC2544, Throughput	pvp_tput	Bi-Directional Multistream=False	Hardware Only	Figs: 21, 22

## Additional Test Configuration Information

Category	Details
Environment of the DUT	OS: CentOS Linux 7 Core Kernel Version: 3.10.0-327.36.3.el7.x86_64 - NIC(s): 5 - Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01) Board: Intel Corporation S2600WT2R [2 sockets] CPU: Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz CPU cores: 88 Memory: 65687516 kB Hugepages: 1G, 32 pages. iommu-enabled.
Environment of the Traffic generators	OS: CentOS Linux 7 Core Kernel Version: 3.10.0-327.36.3.el7.x86_64 - NIC(s): 5 - Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01) Board: Intel Corporation S2600WT2R [2 sockets] CPU: Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz CPU cores: 44 Memory: 65687516 kB Hugepages: 1G, 32 pages. iommu-enabled.
VSPERF	GIT tag: 0498b5fb3893e4331191808e3501d00b684af9b4
OVS	OvsDpdkVhost, Version: 2.6.90, GIT tag: ed26e3ea9995ba632e681d5990af5ee9814f650e
VPP	VppDpdkVhost, Version: v17.01-release

DPDK	Version: 16.07.0, GIT tag: 20e2b6eba13d9eb61b23ea75f09f2aa966fa6325		
Hardware traffic Generator Configuration	Traffic ports: 10G		
BM-A	Version: v0.33 CPU-Pinning: YES Hugepages: DPDK: 2.2.0 Cores/Interface: 1 Txqueues/port:1 (max 128) RxQueues/port:1 (max 128)		
BM-B Details	CPU-Pinning: YES RxQueues/Port:1 TxQueues/Port:2 Hugepages: DPDK: 2.2.0		
Traffic-Generator as a VM	Version: 4.73 RxQueues/Port: 1 VM Configurations: 2-Port, 6 vCPUs. CPU-Pinning: YES. 1:1 Other Details:		
VLOOP VNF	QemuDpdkVhostUser, Version:2.5.0, GIT tag: a8c40fa2d667e585382080db36ac44e216b37a1c Image: vloop-vnf-ubuntu-14.04_20160823.qcow2 Loopback apps: testpmd, Version: 16.07.0, (GIT tag: 20e2b6eba13d9eb61b23ea75f09f2aa966fa6325)		
Noisy-VM	A Stressor-VM Instances: 2		
Noise-levels	Noiselevel-0 CPU-100% Network Load-NIL Readsize: 32K Writesize:32K BufferSize:64K Read-Rade:5000 Write-rate:2000	Noise-Level-1 CPU-100% Network Load – NIL Read-Size: 4MB Write-Size:4MB Buffer-Size: 33MB Read-rate:10000 Write-rate: 5000	Noise-Level-2 CPU-100% Network-load - NIL Read-Size: 8MB Write-Size: 8MB Buffer-Size: 64MB Read-rate:10000 Write-rate:5000

## Analysis of Test Results

Test/Test-Category	Inference	Ref. Figures	Reasoning Explanation Justification
Hardware Traffic Generator	For 64-Bytes, with the hardware traffic generator, and for both OVS and VPP, the maximum forwarding achieved is only 80% of line-rate.	Fig-3	Inherent limitations - memory bandwidth or PCI bandwidth or transactions per second on the PCI bus.
	For single flow, Avg. latency for OVS and VPP varies from 10-90us with minimal (1-9%) difference between them. Average latency jumps significantly after 128 B.	Fig-4	The packet-processing architecture could be one possible reason for the differences.  Note: One should take into consideration of the inconsistencies in latency measurements - along with lack of delay-variations information.

	<p>With multistream (4096 flows and 1M Flows) the throughput performance of OVS is lower compared to VPP for lesser packet sizes (64 and 128).</p> <p>*Inconsistency*</p> <p>•OVS: 4K flows lower TPUT vs 1M</p>	Fig-5 and Fig-7	<p>OVS could be doing more processing on the packets compared to the VPP. Other reasons could be:</p> <ul style="list-style-type: none"> <li>• Difference in packet-handling architectures</li> <li>• Packet construction variation. Results are use-case dependent</li> <li>•Topology and encapsulation impact workloads under-the-hood</li> <li>•Realistic and more complex tests (beyond L2) may impact results significantly</li> <li>•Measurement methods (searching for max) may impact results</li> <li>•DUT always has multiple configuration dimensions</li> <li>•Hardware and/or software components can limit performance (but this may not be obvious)</li> <li>•Metrics / statistics can be deceiving – without proper considerations to above points!</li> </ul>
	<p>For multi-stream, latency variation are:</p> <ul style="list-style-type: none"> <li>•Min: 2-30us</li> <li>•Avg: 5-110us</li> </ul> <p>Inconsistency for 256B with OVS vs VPP.</p> <p>With Multistream (4096 flows and 1M flows) the latency of VPP is higher compared to OVS for lesser packet sizes (64, 128 and 256).</p>	Fig-6 and Fig-8	<p>The inconsistencies should be taken into consideration before making any conclusions.</p> <p>The packet-processing architecture could be one possible reason for the differences.</p>
BM-A as Baremetal Traffic Generator	For Single flow, The throughput achieved with BM-A for both OVS and VPP is consistent with the hardware traffic generators	Figs: 9 and 3.	We should be take into consideration of the inconsistencies with lower-packet sizes. However, for larger packet size software generators can match hardware counterpart for this scenario.
	For multistream, the throughput performance difference between OVS and VPP is lesser (approximately 50%, 30% and 0% for 64bytes, 128 bytes and 256 bytes respectively) compared to hardware traffic generator (approximately 70%, 50% and 5%)	Figs: 11 and 5	<p>Possible reasons:</p> <ul style="list-style-type: none"> <li>• Packet construction variations</li> <li>• Difference in test traffic type.</li> </ul>
	There are issues with latency computations in BM-A.	Figs 12 and 14.	Resource requirement for latency measurements are not satisfied by the configurations - Work to fix this issue is in progress.
BM-B as Baremetal Traffic Generator	The throughput results for VPP, with BM-B is consistent with the hardware traffic generator. It is able to send and receive at the same rate as hardware traffic generator	Fig 15 and Fig3.  Fig 16 and Fig 6.	We should be take into consideration of the inconsistencies with lower-packet sizes. However, for larger packet size software generators can match hardware counterpart for this scenario.
	OVS throughput performance with BM-B is totally different – lesser for single flow vs VPP, and same as VPP for multistream – in comparison with other traffic generators.	Fig 15	The way the packet is formed by BM-B could be one possible reason.
	BM-B yet to support latency measurements in VSPERF	N/A	The work is in progress to support latency measurements.
Traffic generator as VM.	Traffic generator as VM is yet to match the traffic generation capabilities of Baremetal traffic-generators. The throughput for 64-bytes is almost 50% that of BM-A or BM-B.	Figs: 17 and 19	<p>Inherent overhead of running in VM vs running baremetal could be one possible reason. This could be confirmed if we run BM-A BM-B in a VM.</p> <p>TGen-VM may be doing extra work per packet.</p> <p>TGen-VM is doing both port-level statistics and stream-level statistics and it may not apply to others.</p> <p>TGen-VM with improved configuration - multiple rx-queues and CPU-cores - such as that of BM-A BM-B should improve its performance.</p>
	<p>Among the software generators the latency measures of TGen-VM is consistent with the hardware latency measurements.</p> <p>However, the latency values (min and avg) can be 10x times the values provided by the hardware traffic-generator.</p>	Figs: 18 and 20.	<p>TGen-VM does port level statistics and stream level statistics including checking for packet sequencing errors and dropped packets, etc.</p> <p>Possible reason for high/inconsistent latency measurements could be Configuration of NTP servers.</p>
	With TG as a VM there was no 'significant' performance difference between OVS and VPP – neither the throughput nor the latency,	Figs 17, 18, 19 and 20.	Possibly due to the lower-throughput rate. At that rate, both the switches should be able to perform similarly.

	The average latency significantly increases for larger-packet sizes for both single flow and 4096Flows.	Figs 18 and 20.	This trend is seen for all cases.
Comparison of Baremetal Traffic generators.	There is lack of consistency of OVS throughput results between the two traffic generators for both Single flow and multi-stream. However, with VPP, they are very consistent.	Figs: 9 and 15.  And, Figures 11 and 16	It needs additional experiments to understand and explain this aspect.
	Both the traffic generators lack proper latency measurements – as of now.	Fig 10 and 12.	Latency measurements are expected to improve by next-release.
Virtual Switches	VPP can forward packets at a faster rate compared to OVS for multi-stream case and for lower-packet size. The %ge of difference can range from 80% to 30%.	Figs 5, 7, 11, 13, and 15.	Packet-handling architectures: The amount of processing per-packet: VPP may be lesser compared by OVS.  Configuration of the OVS needs to be analysed in detail to explain the performance difference.  There still exists many inconsistencies and results differ across traffic generators.
	The cache miss of VPP is 6% lower compared to the cache-misses for OVS.	Fig 26.	Processing architecture of VPP could be one possible reason.
	For both OVS and VPP the latency jumps significantly for larger packet-sizes.	Figs: 4, 6, 8, 18 and 20.	A thorough analysis of latency measurement is required.
Noisy Neighbor Tests	The performance (throughput and latency) with PVP topology is extremely poor compared to P2P topology.	Figs: 22 and Fig 22.	The L2FWD module configuration and virtual-interface could be possible reason for degradation in performance.
	If a noisy neighbor consumes the L3 cache fully and overload the memory bus (noiselevel 1 and 2), it can create sufficient noise and degrade the performance of the VNF.	Fig 21.	The Last-level cache is a shared resource - among all CPUs. Hence, a single process consuming L3-cache completely will affect the performance other processes.  CPU affinity configuration and NUMA configuration can protect from majority of Noise.  Consumption of Last-level cache (L3) is key to creating noise.  It maybe be worth studying the use of tools such as cache-allocation-technology (Libpqos) to manage noisy-neighbors
	The effect on latency due to noisy-neighbors is minimal	Fig-22	The low-throughput performance could be a reason for this trend.
Hardware Vs Software Traffic generators.	Software Traffic Generators on bare-metal are comparable to HW reference for larger pkt sizes. Small pkt sizes show inconsistent results •Across different generators •Between VPP and OVS •For both single and multi-stream scenarios	Fig 27.	TG characteristics can impact measurements  •Inconsistent results seen for small packet sizes across TGs •Packet stream characteristics may impact results ... bursty traffic is more realistic! •Back2Back tests confirm sensitivity of DUT at small frame sizes •Switching technology (DUT) are not equally sensitive to packet stream characteristics. Configuration of 'environment' for Software traffic-generators is critical - such as  1. CPUs: Count and affinity definition 2. Memory: RAM, Hugepages and NUMA Configuration 3. DPDK Interfaces: Tx/Rx Queues 4. PCI Passthrough or SRIOV configurations 5. Software version

## Test Topology

Intel POD 12 was dedicated for the event, which had 6 servers with normal Pharos configuration. The important nodes of the configuration are

1. pod12-node 4 – As shown on left-top side of Figure-1
  - a. SUT (Sandbox) – Vserver with either OVS or VPP.
  - b. Stress-VM
  
1. pod12-node5 – Shown on the right side of the Figure-1
  - a. SW traffic generators
    - i. BM-B : Use interfaces eno3 and eno4 through DPDK.
    - ii. BM-A-trafficgen: Use interfaces eno3 and eno4 through DPDK
    - iii. TGen-VM with Passthrough: Uses the itnerfaces eno3 and eno4 via passthrough and DPDK.
  
1. Hardware Traffic Generator - Chassis – Shown in left-bottom side of the Figure-1

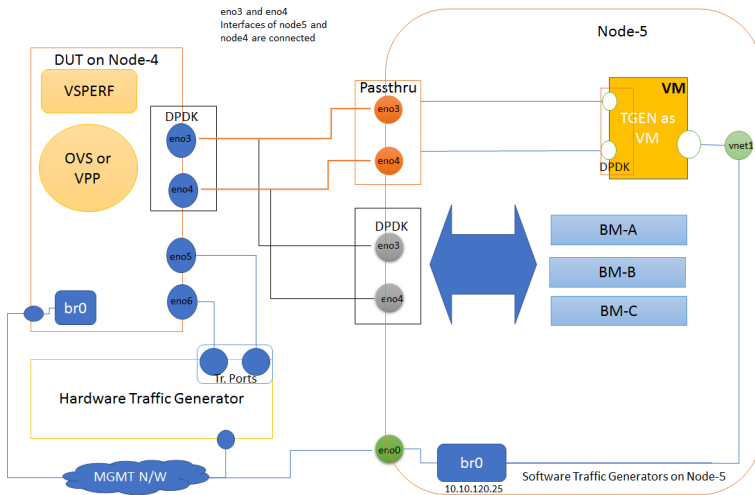


Figure 1: Test Topology

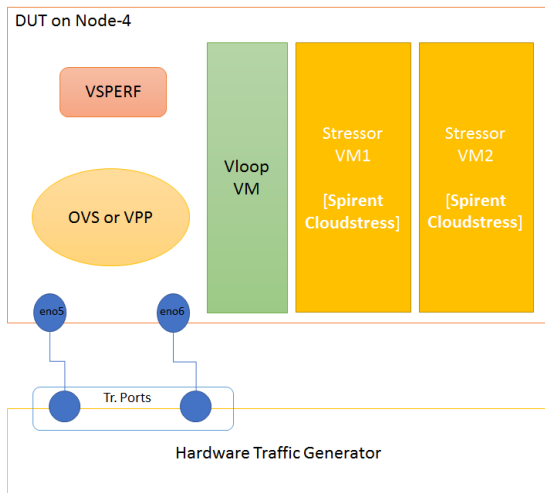


Figure 2: Test Topology used for Noisy-neighbor tests

## Test Results

### Hardware Traffic Generator Results

#### RFC2544 OVS and VPP – Single Flow, Bidirectional

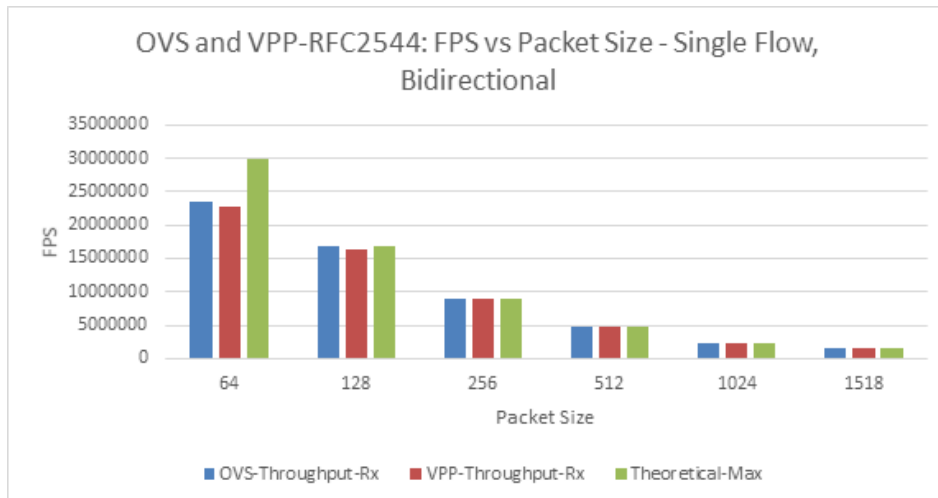


Figure 3: OVS and VPP - RFC2544: FPS Vs Packet Size

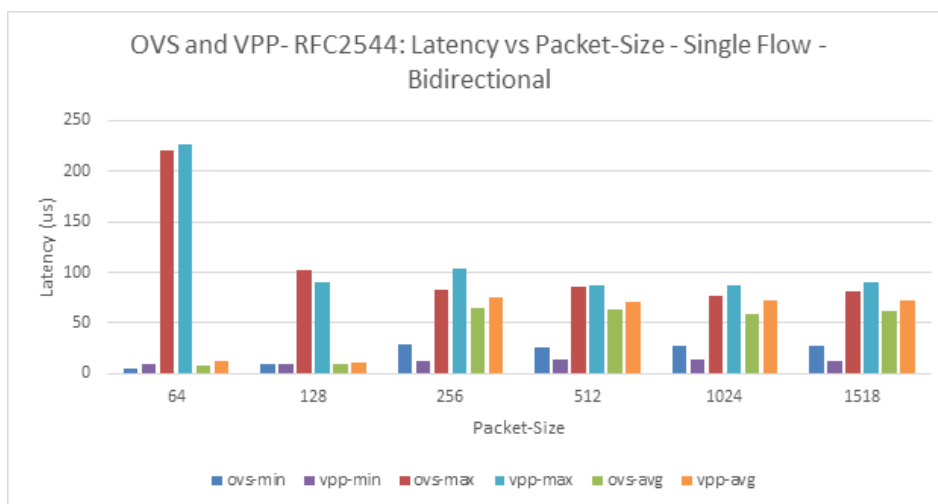


Figure 4: OVS and VPP- RFC2544: Latency vs Packet Size – Single Flow

## RFC2544 OVS and VPP Multi-Stream.

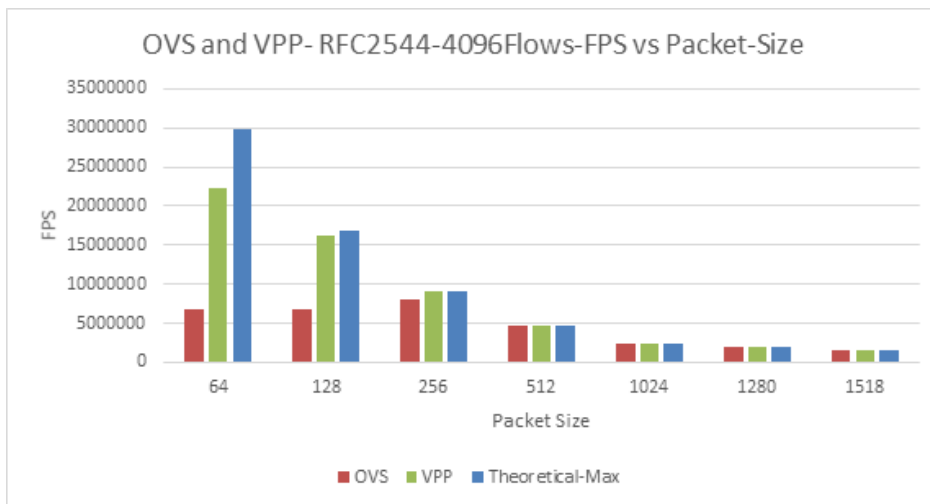


Figure 5: Ovs and VPP, RFC2544-4096 Flows- FPS vs Packet-Size

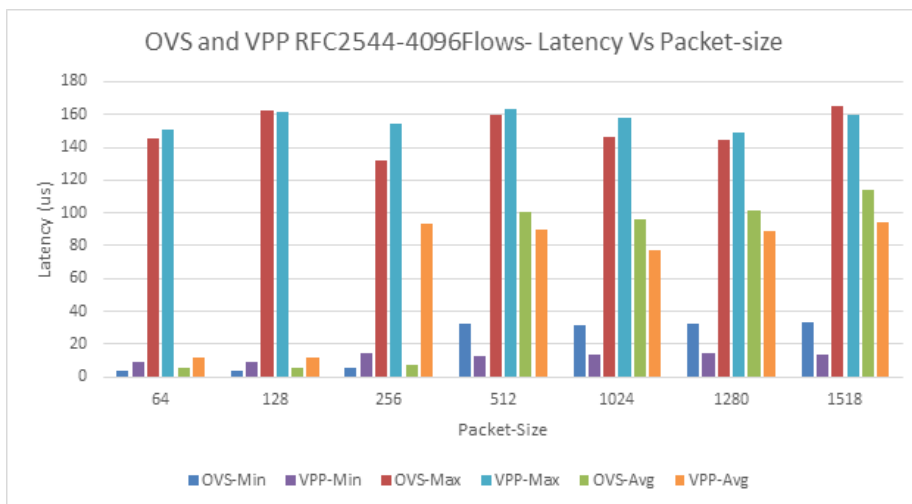


Figure 6: OVS and VPP, RFC2544 4096Flows: Latency Vs Packet-Size

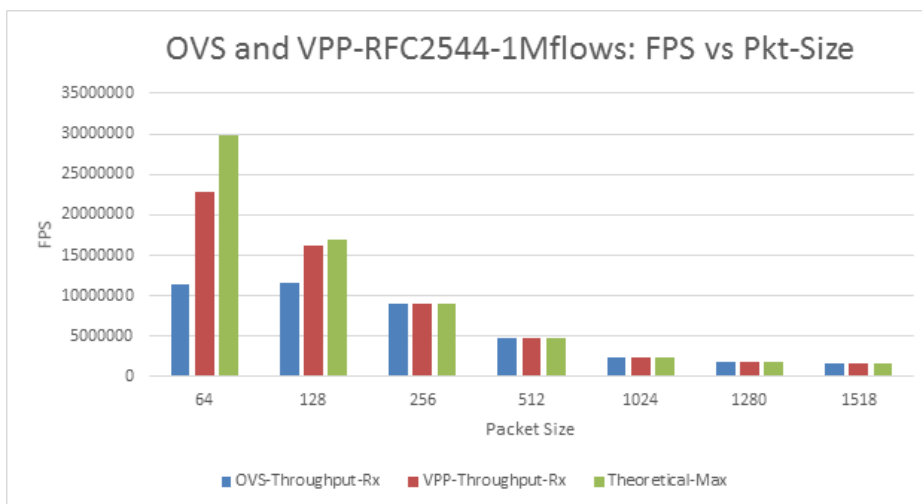


Figure 7: OVS and VPP 1M Flows FPS vs Packet-Size



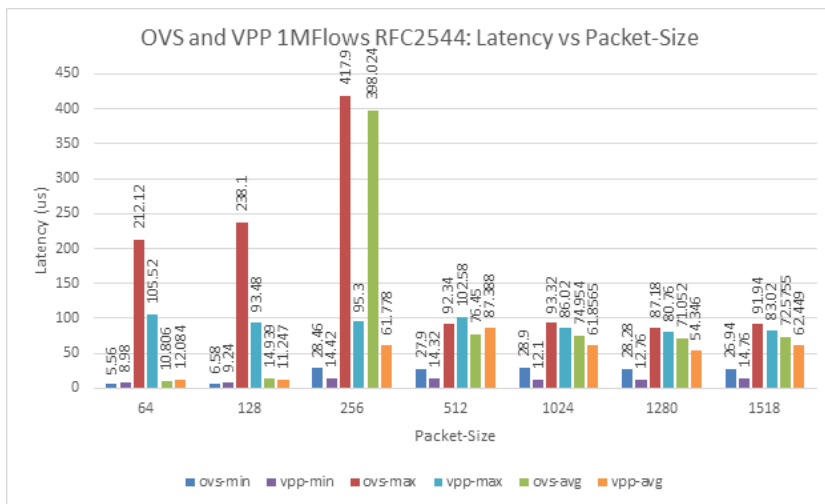


Figure 8: OVS and VPP - 1M Flows. Latency Vs Packet-Size

## Baremetal (BM-A) Results

### RFC2544 OVS and VPP – Single Flow, Bidirectional

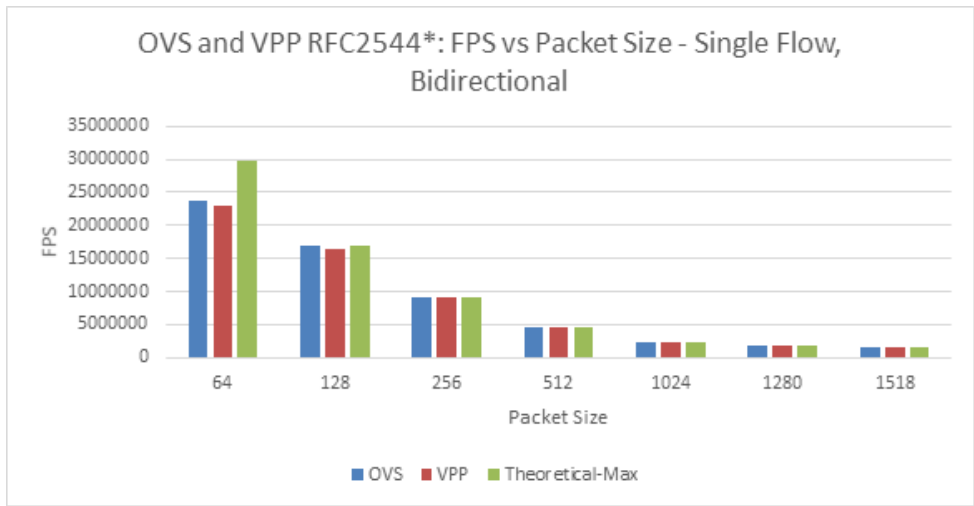


Figure 9: BM-A-OVS and VPP Single Flow -FPS vs Packet-Size

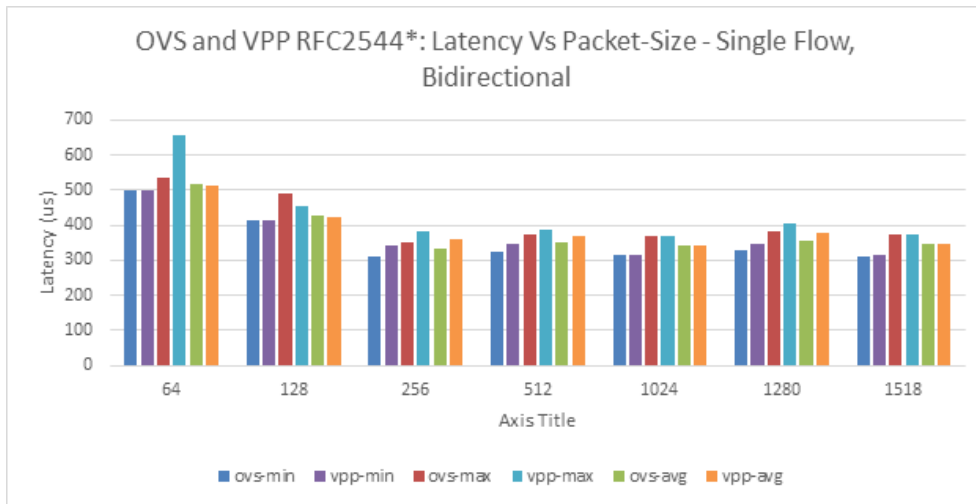


Figure 10: BM-A-OVS and VPP Single Flow - Latency Vs Packet-Size

### RFC2544 OVS and VPP Multi-Stream.

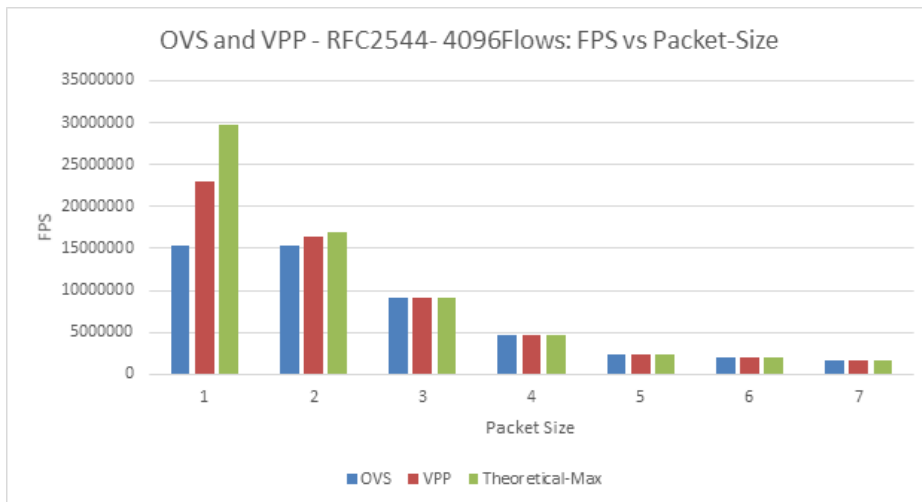


Figure 11: BM-A - OVS and VPP, 4096Flows: FPS vs Packet-Size

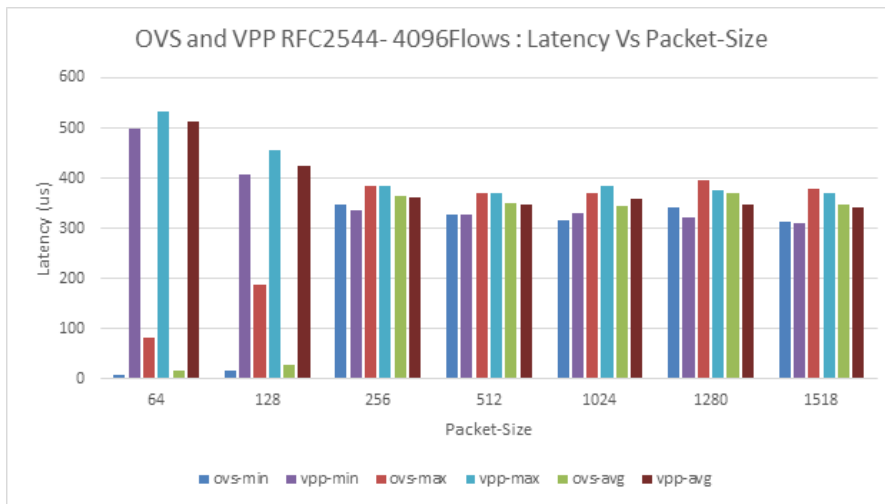


Figure 12: BM-A - OVS and VPP, 4096Flows: Latency Vs Packet-Size

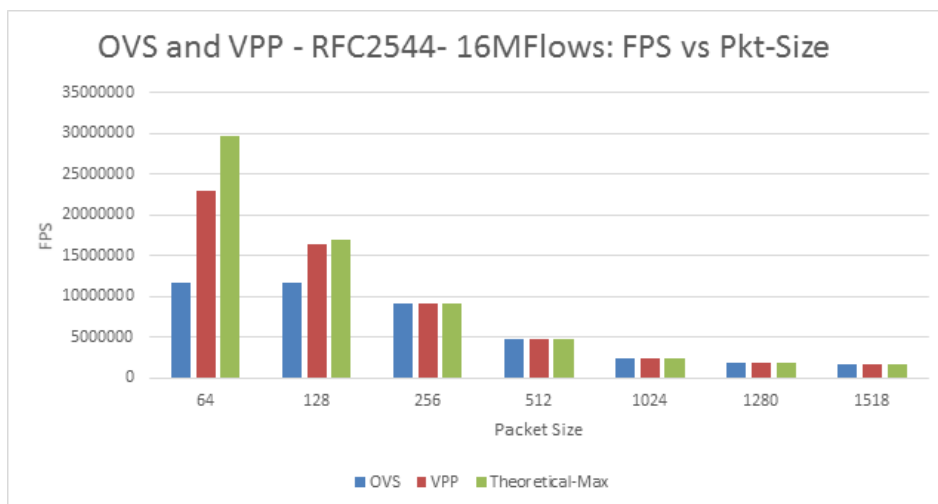


Figure 13: BM-A - OVS and VPP, 16M Flows : FPS vs Packet-Size

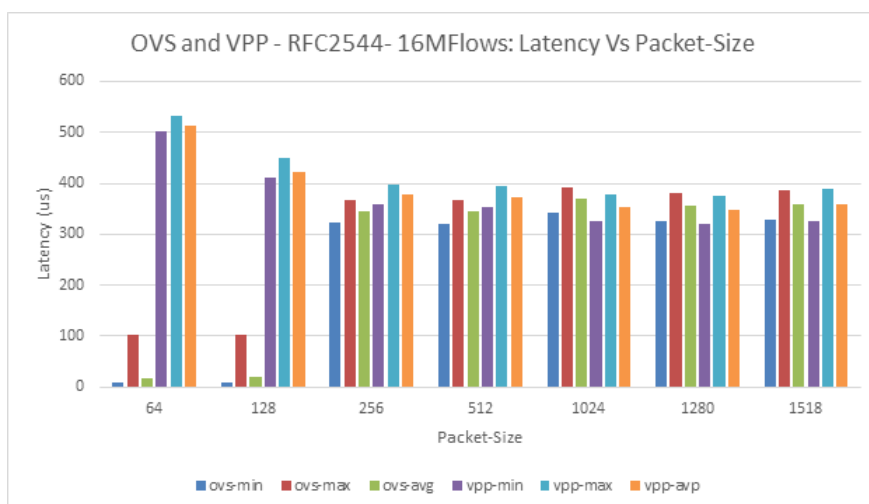


Figure 14: BM-A - OVS and VPP, 16M Flows : Latency Vs Packet-Size

## Baremetal (BM-B) Results

### RFC2544 OVS and VPP – Single Flow, Bidirectional

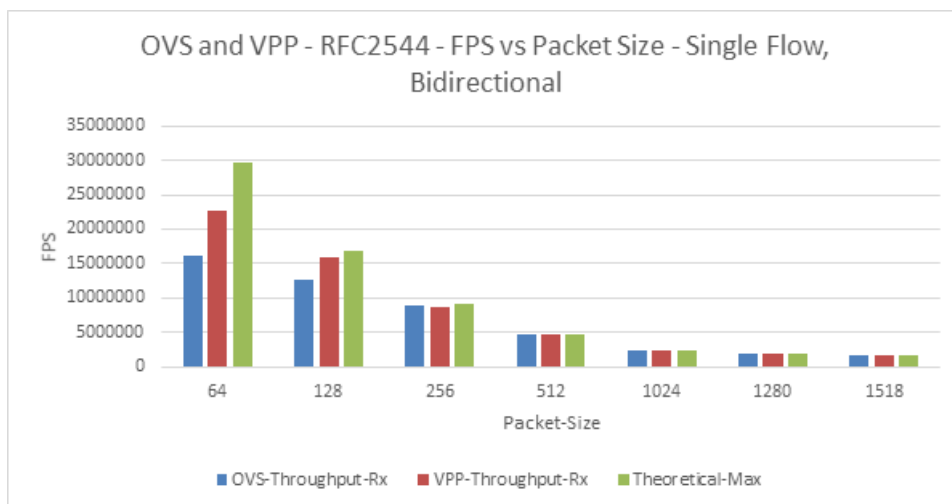


Figure 15: BM-B - OVS and VPP – Single Flow - FPS vs Packet Size

### RFC2544 OVS and VPP Multi-Stream.

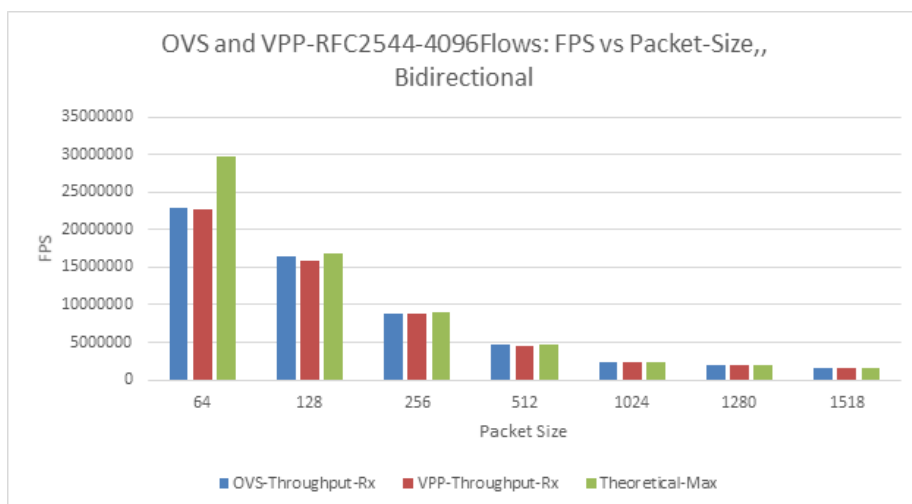


Figure 16: BM-B- OVS and VPP, 4096 Flows : FPS vs Packet-Size

### Traffic Generator as VM Results

#### RFC2544 OVS and VPP – Single Flow, Bidirectional

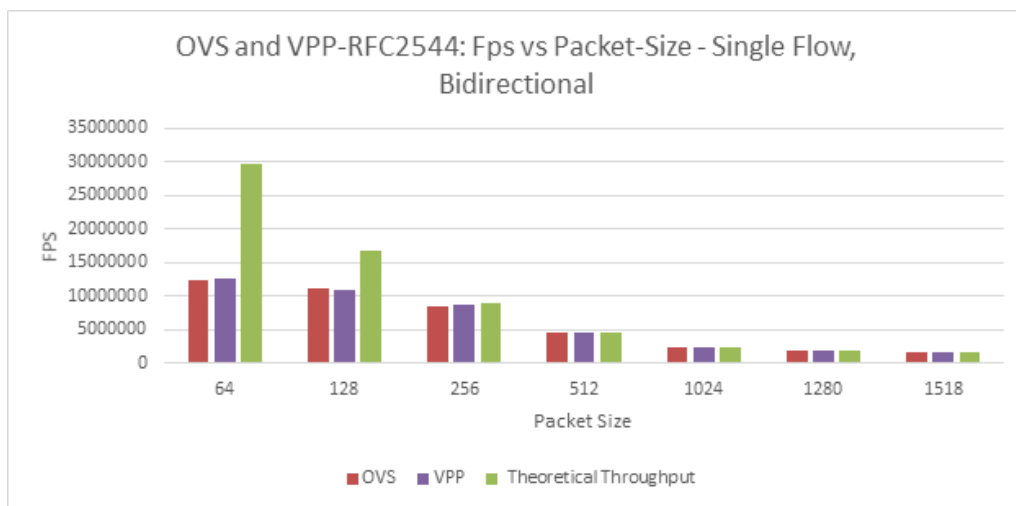


Figure 17: OVS and VPP Single Flow FPS vs Packet-Size

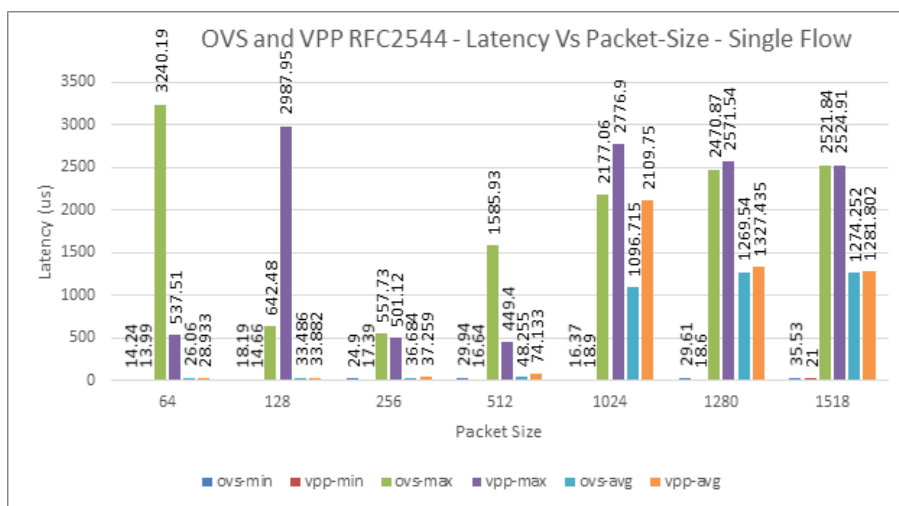


Figure 18: OVS and VPP, Single Flow - Latency Vs Packet-Size

### RFC2544 OVS and VPP Multi-Stream.

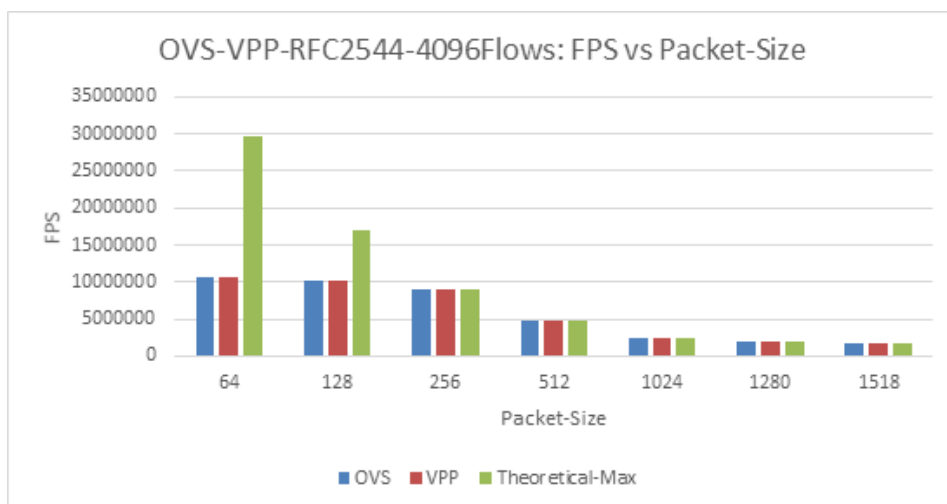


Figure 19: OVS and VPP, 4096 Flows: FPS vs Packet-Size

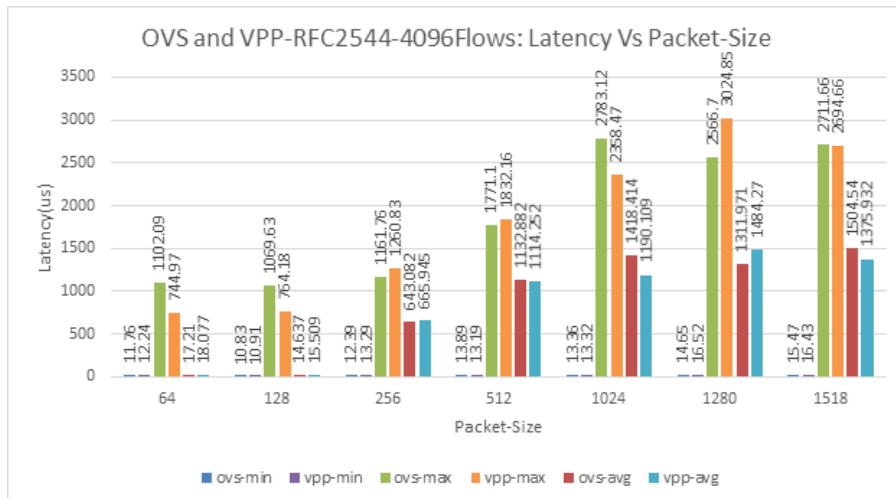


Figure 20: OVS and VPP, 4096Flows Flows - Latency vs Packet-Sie

## Noisy-Nighbor test results

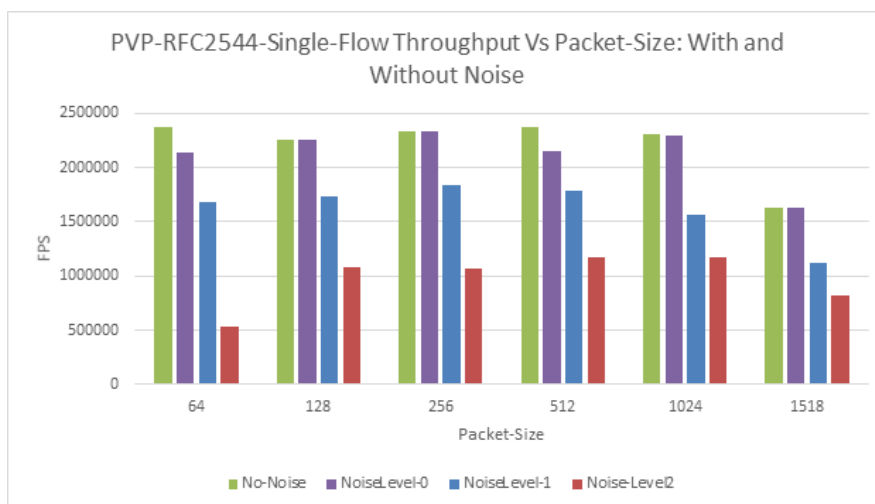


Figure 21: PVP-RFC2544 Single-flow - FPS vs Packet Size - With and Without Noise

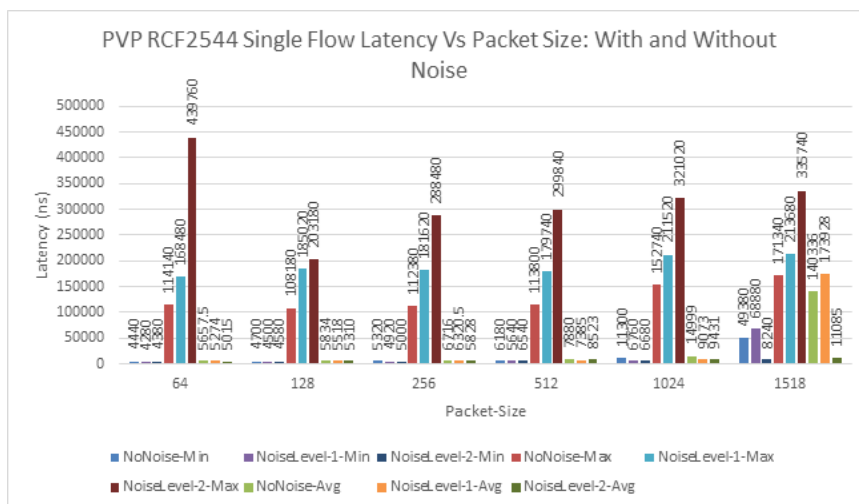


Figure 22: PVP, RFC2544 Single Flow, Latency Vs Packet Size: With and Without Noise

## Future Works.

VSPERF team plans to complete the following tests before next Plugfest.

Category	Explanation
Study of Consistency of results of the traffic generators.	
Integration of Other software traffic generators	
In-depth Analysis of OVS	
In-depth Analysis of VPP	

## Appendix-A: BM-B Re-Run

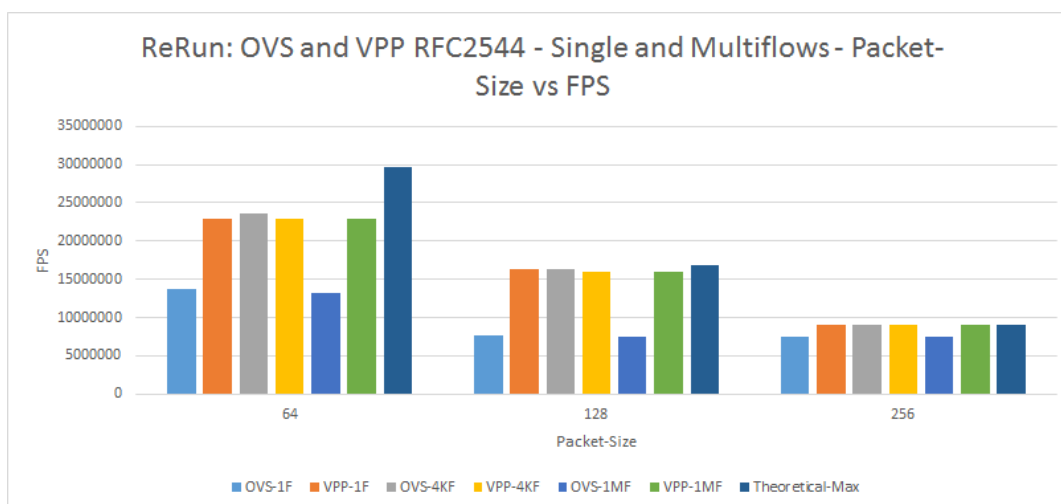


Figure 23: OVS and VPP RFC2544 - Single and Multiflows - Packet-Size vs FPS

# Appendix B: Back2Back Testing Time Series (from CI)

[RFC2544 Back-to-Back tests](#) attempt to determine the maximum length of a stream of frames (sent with minimal spacing, or back-to-back) that can be transmitted through the DUT without loss.

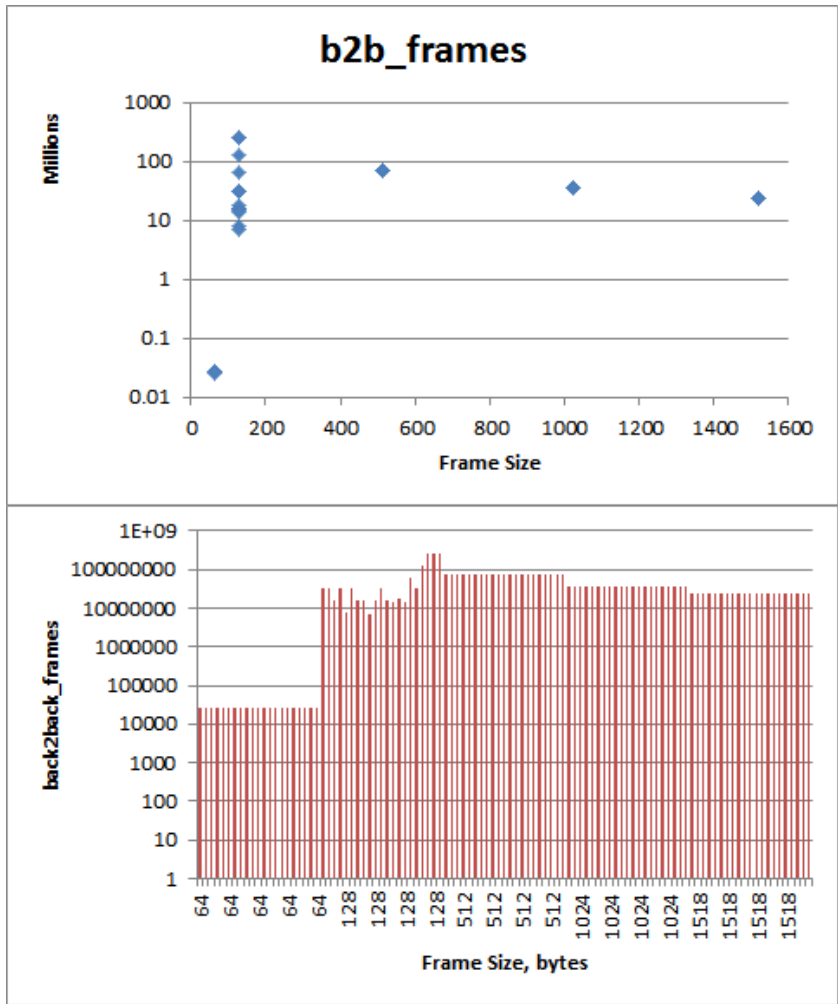


Figure B-1: OVS RFC 2544 Back-to-Back CI Testing of P2P Deployment on POD12 in 2017 (individual measurement time series on the right)

The [Purpose of back to back testing](#) is to assess the amount of data buffering in a networking device or function. There are (at least) 3 properties of the test set-up which influence the results:

- 1. The Max Header Processing Rate of the DUT
- 2. The effective length of **all** buffers the frames encounter as they pass through the DUT
- 3. The characteristics of the stream of frames and Traffic Generator that controls them.

A simple model of the test setup includes the following functions in sequence:

Frame Generator -> Buffer -> Frame Processor -> Test Receiver

If the Max Header Processing Rate of the DUT is less than the rate of Frame Generation, then buffering will occur, and potentially frame loss.

The VSPERF Back2Back test uses five different frame sizes, and each frame size has a **Maximum Frame Rate** that depends on the **minimum Inter-Frame Gap (12 bytes)**, the **Ethernet Preamble (8 bytes)**, and the **nominal interface bit rate (10 Gigabits/second on eno5 and eno6 of Node 4)**.

Table B-1 Back2Back Frame measurement results by Frame Size, Compared with Theoretical and Measured Frame Rates

Packet size, bytes	64	128	512	1024	1518
Max Frame Rate @10Gb/s (Theoretical)	14,880,952	8,445,945	2,349,624	1,197,318	812,743
RFC 2544 Throughput (One-way), FPS (Ave of 20 CI results, Pod 12)	11,696,726	8,340,147	2,349,596	1,197,301	812,732
Ave Back2Back Frames Measured	26,700	(Max) 2.53E+08	70,488,721	35,919,540	24,382,314



Ave Implied Buffer Time, seconds	0.001794	(Max) 30	30	30	30
----------------------------------	----------	----------	----	----	----

From Table B-1, it is clear that only the 64 byte frame size has a Max Frame Rate sufficient to reliably cause Buffering and Loss. For larger Frame Sizes, the average Throughput measured is sufficient to handle the Max Frame Rate (Note: Measured Throughput averages include some sub-par measurements, and the Averages are slightly lower than the Max.) Observe that the large number of Back2Back Frames reported represent 30 seconds of buffer time (impossible) and are more likely an artifact of the test duration specified at the Frame Generator. Figure B-2 Illustrates the estimated Buffer Times, and consistency across the 20 CI test results for each frame size can be evaluated.

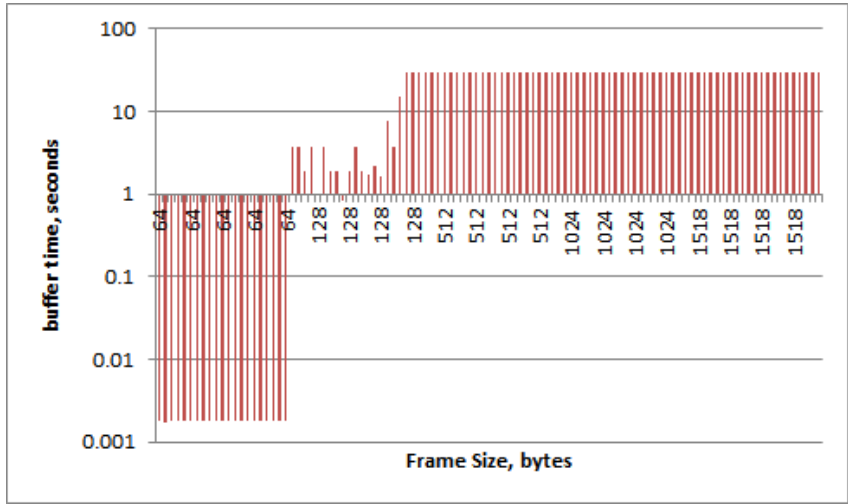


Figure B-2 Estimated Buffer Time for OVS RFC 2544 Back-to-Back CI Testing of P2P Deployment on POD12 in 2017

A **Source of Error** for the only useful estimate of Buffer Time (64 byte) can be traced to the **number of frames removed from the Buffer (by Frame Processing)** before a frame loss occurs due to overflow. As Frames arrive at Max Frame Rate, a portion of frames are removed from the Buffer according to the ratio between the Measured Throughput and the Max Frame Rate. The remaining Frames represent the **Buffer Time corrected for Frame processing**. We calculate that **only 5,713 Frames remain in the Buffer, corresponding to 0.000384 seconds**. It is important to distinguish this source of error and estimate the actual Buffer size, especially for scenarios where the measured Throughput will not be achieved, or when Frame processing is suspended temporarily.

For a different view of Back2Back measurement consistency, we look at 22 measurements conducted in CI for POD 3 in Figure B-3.

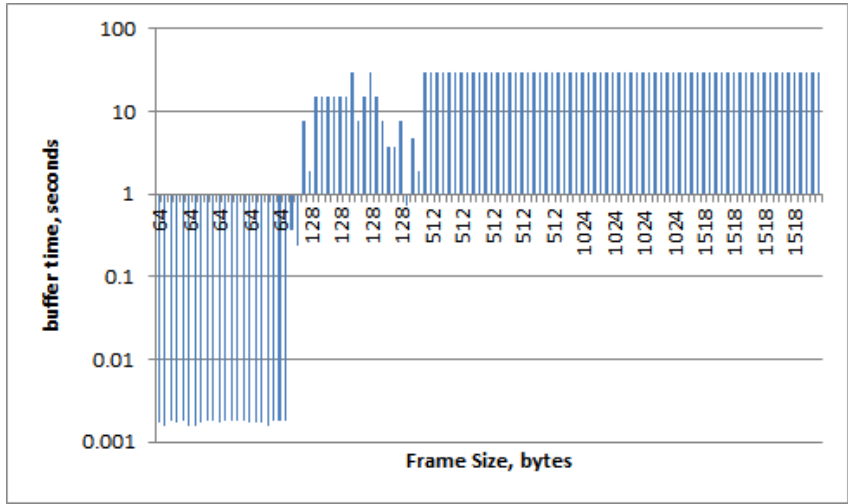


Figure B-3 Estimated Buffer Time for OVS RFC 2544 Back-to-Back CI Testing of P2P Deployment on POD 3 in 2016

POD 3 uses different hardware for the DUT. We can see that the estimated Buffer Times for 64 byte Frames are slightly more variable, with an average of 25796 Frames or 0.001733 seconds (uncorrected for Frame Processing). Again, tests using 128 byte Frames show that the DUT is on the cusp of achieving Max Frame Rate, and yield impossible buffer size estimates.

When we change the deployment scenario in the DUT to PVP, the measured Throughput is greatly reduced and more than one Frame size may produce a useful estimate.

## Appendix C: BM-C Trafficgen Results

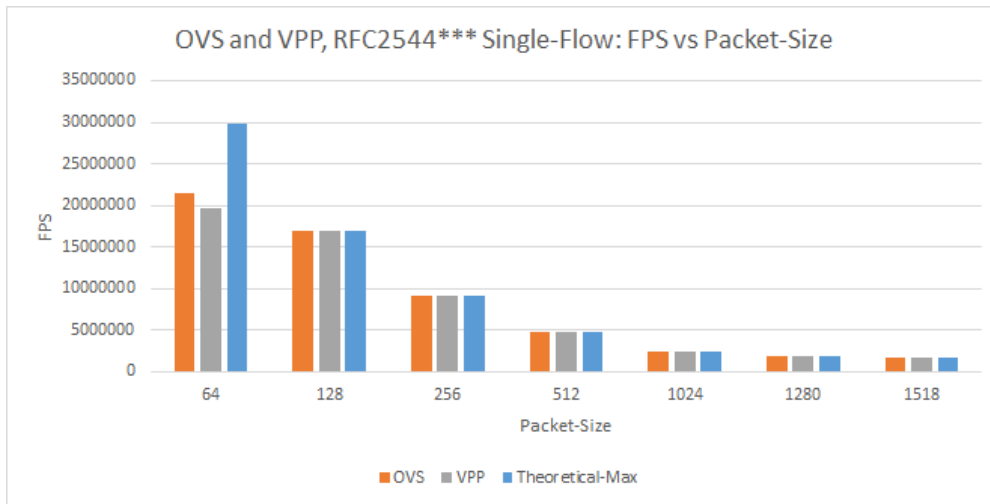


Figure 25: BM-C: OVS and VPP - RFC2544: FPS Vs Packet Size

## Appendix D: Analysis of Cache-Misses

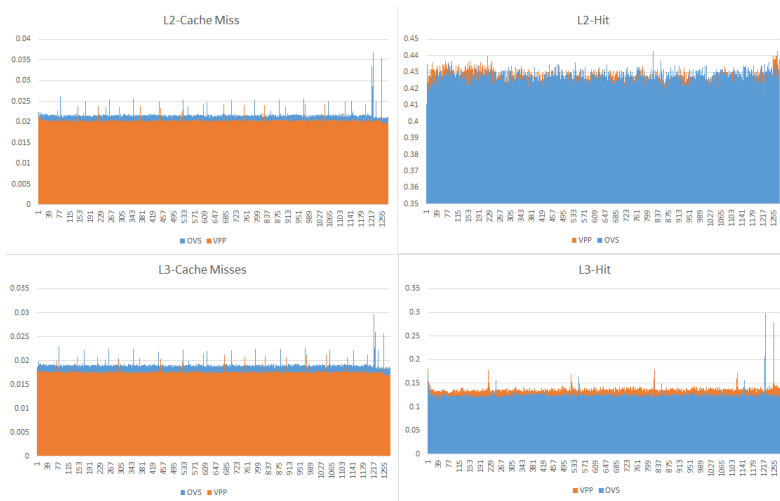


Figure 26: Cache hits and misses for OVS and VPP

## Appendix E: Hardware and Baremetal Traffic Generators - Summarized

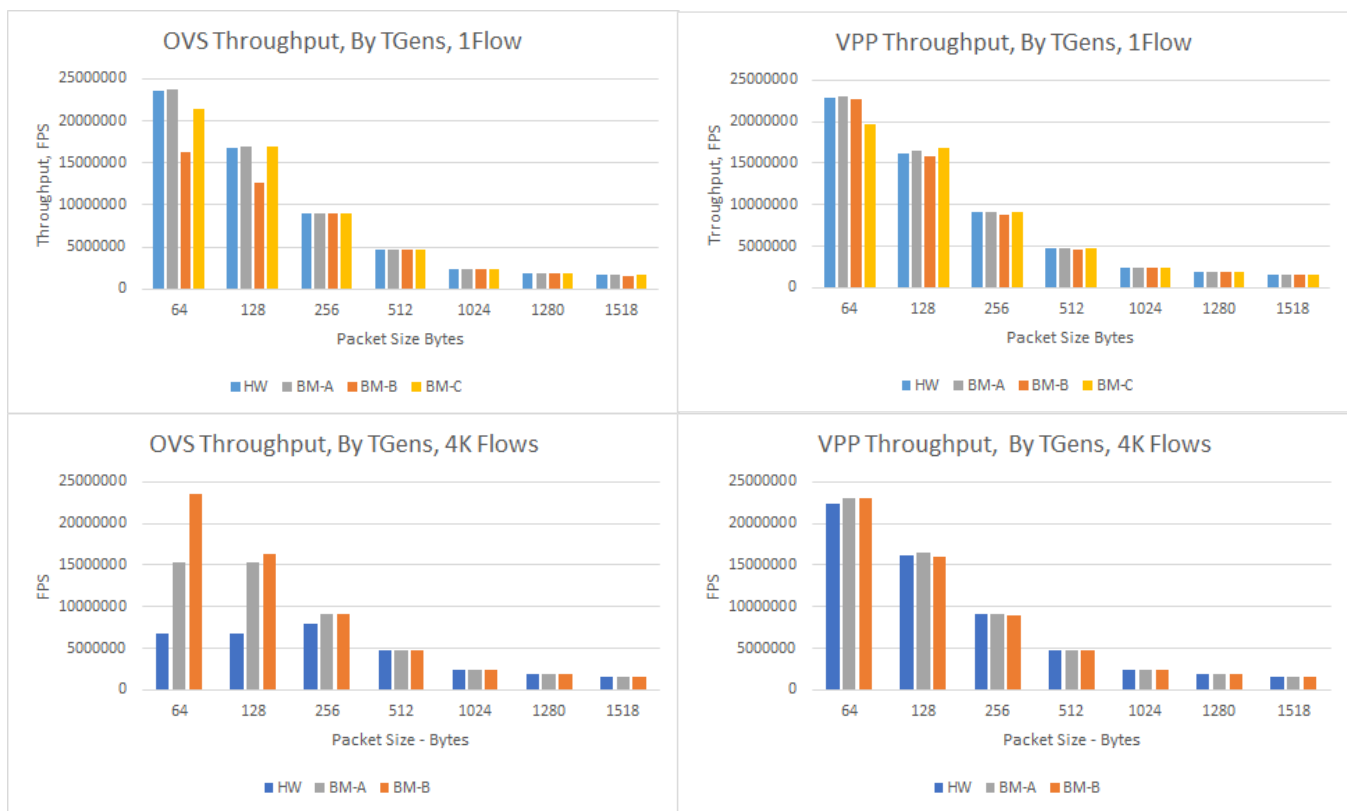


Figure 27: Hardware and Baremetal Summarized.

Note: The TGen configurations for multistream were different, and this may explain some of the performance variation observed.

HW - L2 Dst variation

BM-A - L3 Dst variation (L3 Src variation exhibited higher Tput)

BM-B - L3 Src variation (binary-search.py, where flowMods set to src-ip)