# Cross-NUMA performance measurements with VSPERF



## Introduction:

Cross-NUMA tests as part of OPNFV Plugfest (Gambia) - January 2019, by Sridhar K. N. Rao (sridhar.rao@spirent.com) Al Morton (acmorton@att.com)

1. VSPERF-Scenarios: P2P and PVP.
2. Workloads: vSwitchd, PMDs and VNF.
3. VNF: L2 Forwarding
4. vswitch: OVS and VPP.

## Testcases Run:

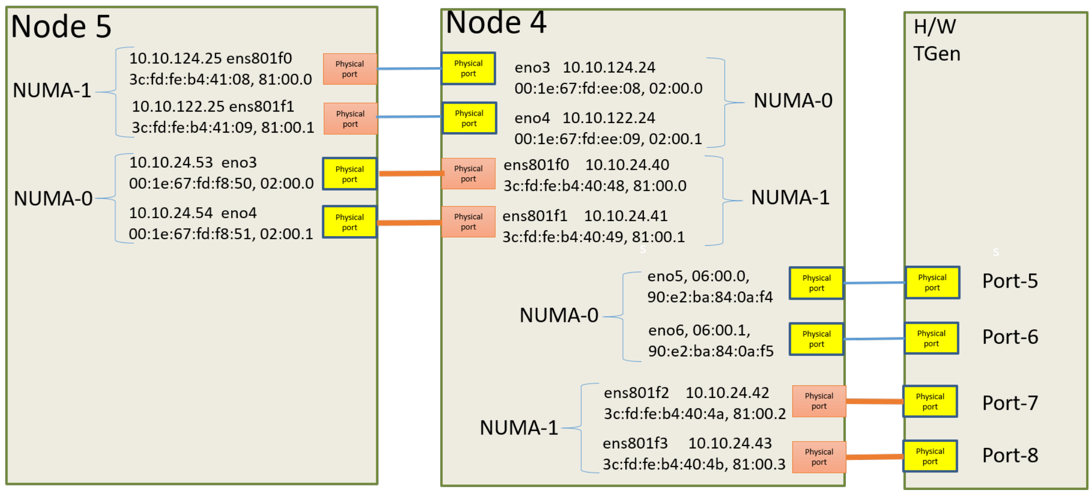
Framesizes: 64, 128, 256, 512, 1024, 1280, 1518

1. RFC2544 Throughput Test - NDR.
2. Continuous traffic Test - 100%

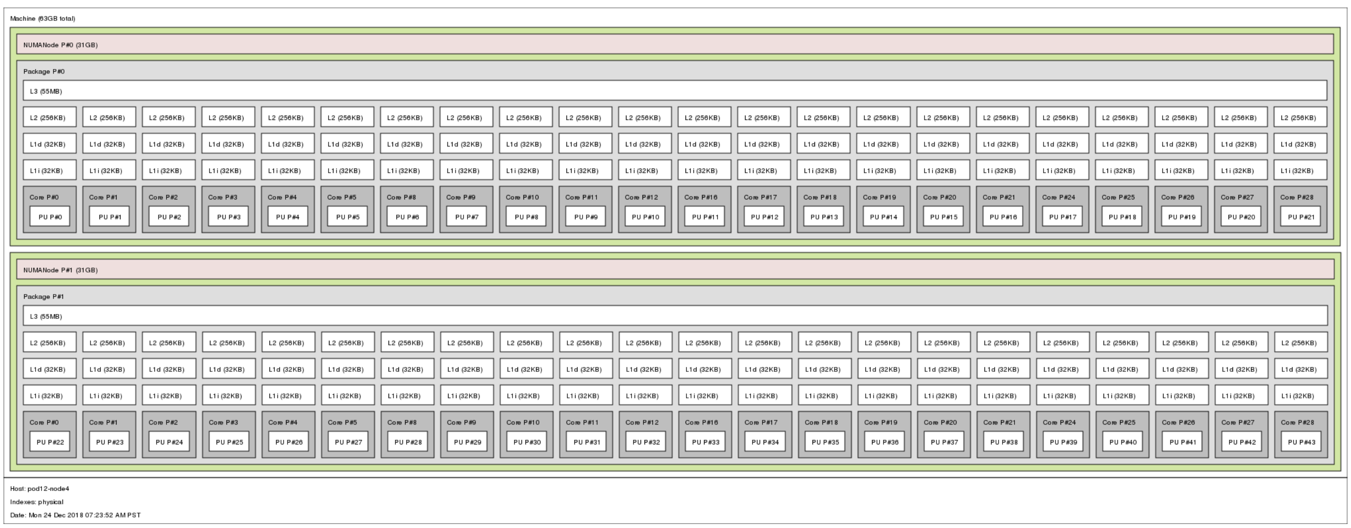
## Testbed:

Intel POD12

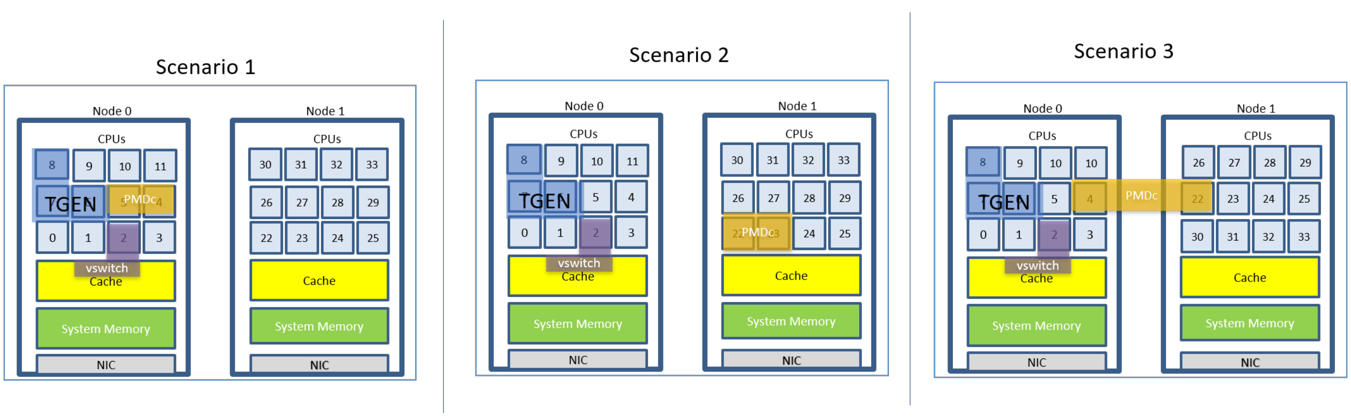
Node-4 (DUT), Node-5 (Software Traffic Generators) and H/W Traffic Generator.




## CPU Topology on DUT



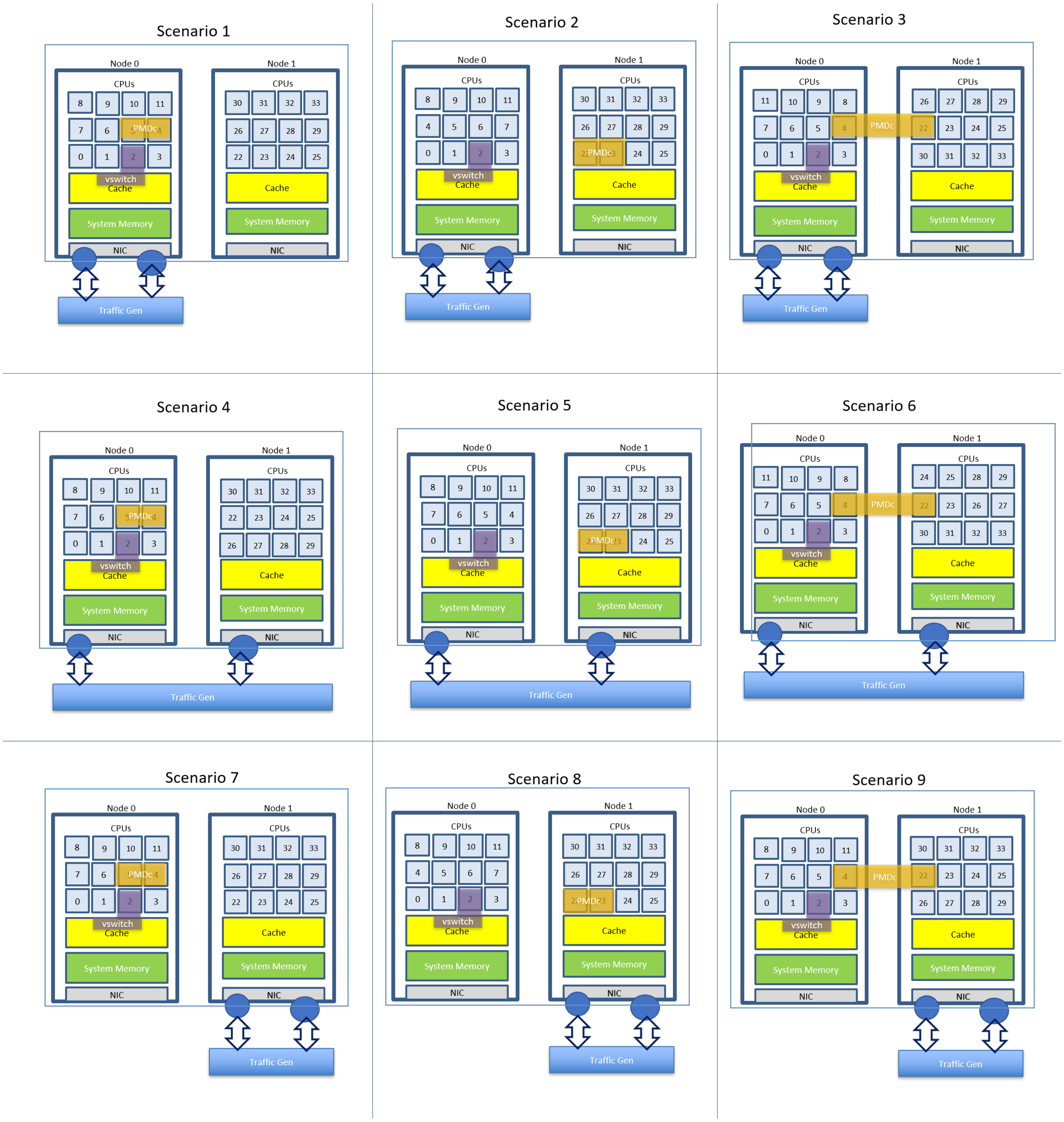

## V2V Scenarios




## Summary of V2V Scenarios

| Scenarios | Possible Core-allocations:<br>Assumptions: Numa-0 (0-21) Numa-1 (22-43)<br><br>vSwitch Core #: 02 | TGen Ports Info |
|---|---|---|
| 1 | PMDs: 4, 5 (0x30) | 2 Virtual Ports 10G |
| 2 | PMDs: 22, 23 (0xC00000) | 2 Virtual Ports 10G |
| 3 | PMDs: 4, 22 (0x400010) | 2 Virtual Ports 10G |

# P2P Scenarios

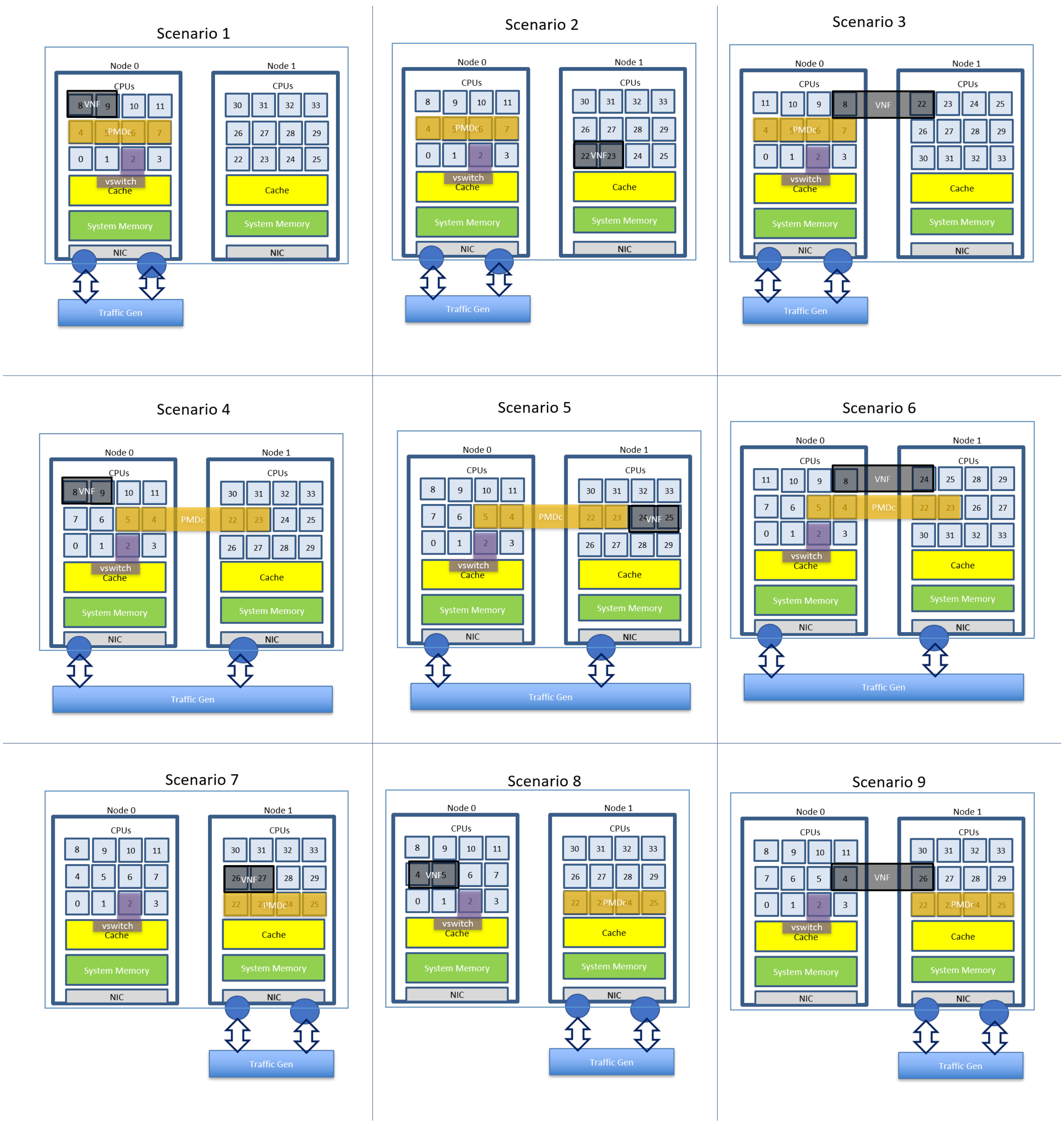



# Summary of P2P Scenarios:

| Scenario | Possible Core-allocations:<br>Assumptions: Numa-0 (0-21) Numa-1 (22-43)<br>vSwitch Core #: 02 | DUT Ports, TGen (Hardware) Ports |
|---|---|---|
| 1 | PMDs: 4, 5 (0x30) | DUT: eno5, eno6<br><br>TGEN: 5, 6 |
| 2 | PMDs: 22, 23 (0xC00000) | DUT: eno5, eno6<br><br>TGEN: 5, 6 |

| | | |
|---|---|---|
| 3 | PMDs: 4, 22 (0x400010) | DUT: eno5, eno6<br>TGEN: 5, 6 |
| 4 | PMDs: 4, 5 (0x30) | DUT: eno5, ens801f2<br>TGEN: 5, 7 |
| 5 | PMDs: 22, 23 (0xC00000) | DUT: eno5, ens801f2<br>TGEN: 5, 7 |
| 6 | PMDs: 4, 22 (0x400010) | DUT: eno5, ens801f2<br>TGEN: 5, 7 |
| 7 | PMDs: 4, 5 (0x30) | DUT: ens801f2, ens802f3<br>TGEN: 7, 8 |
| 8 | PMDs: 22, 23 (0xC00000) | DUT: ens801f2, ens802f3<br>TGEN: 7, 8 |
| 9 | PMDs: 4, 22 (0x400010) | DUT: ens801f2, ens802f3<br>TGEN: 7, 8 |

# PVP Scenarios



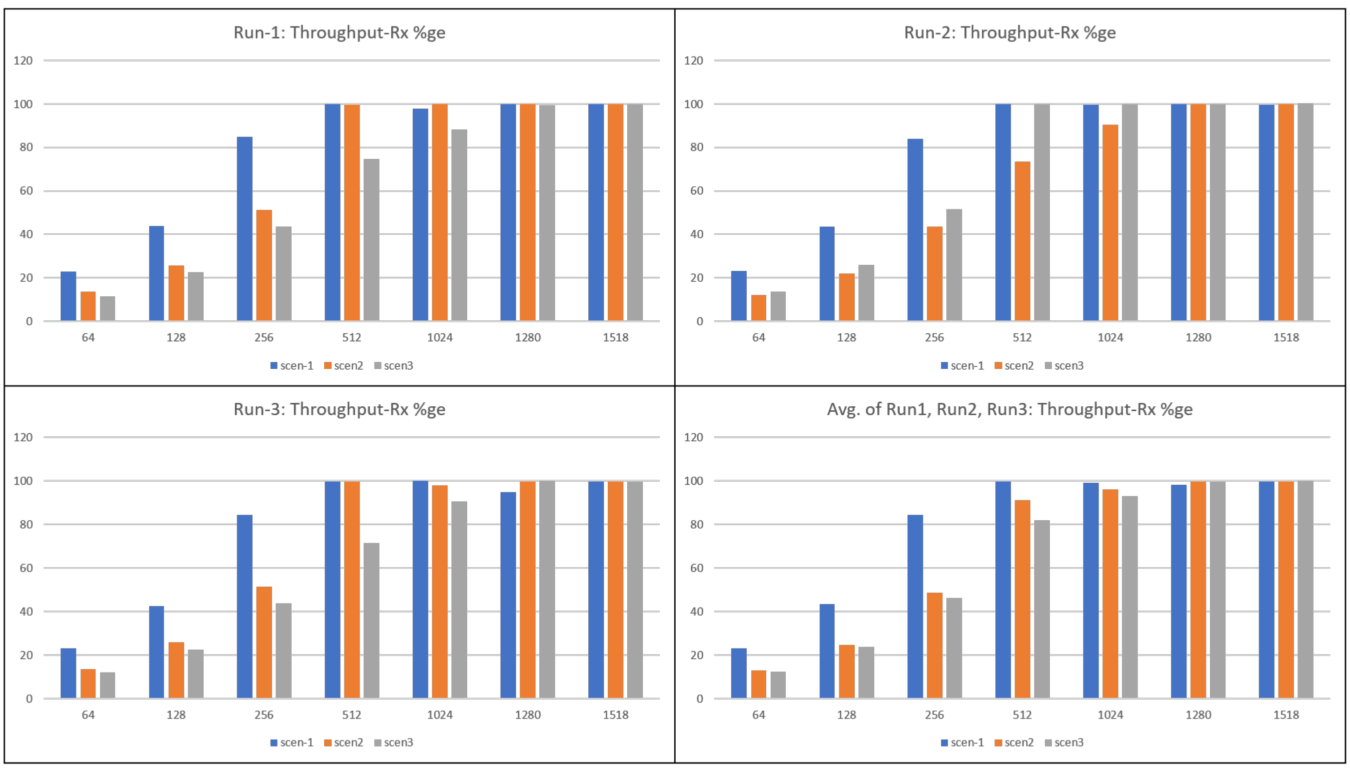

# Summary of PVP Scenarios:

| Scenario | Possible Core-allocations:<br>Assumptions: Numa-0 (0-21) Numa-1 (22-43)<br>vSwitch Core # : 02 | | DUT Ports<br>TGen Ports<br>(Hardware) |
|---|---|---|---|
| 1 | PMDs: 4, 5, 6, 7<br>(0xF0) | VNF: 8,9 | DUT: eno5, eno6<br>TGEN: 5, 6 |
| 2 | PMDs: 4, 5, 6, 7<br>(0xF0) | VNF: 22, 23 | DUT: eno5, eno6<br>TGEN: 5, 6 |

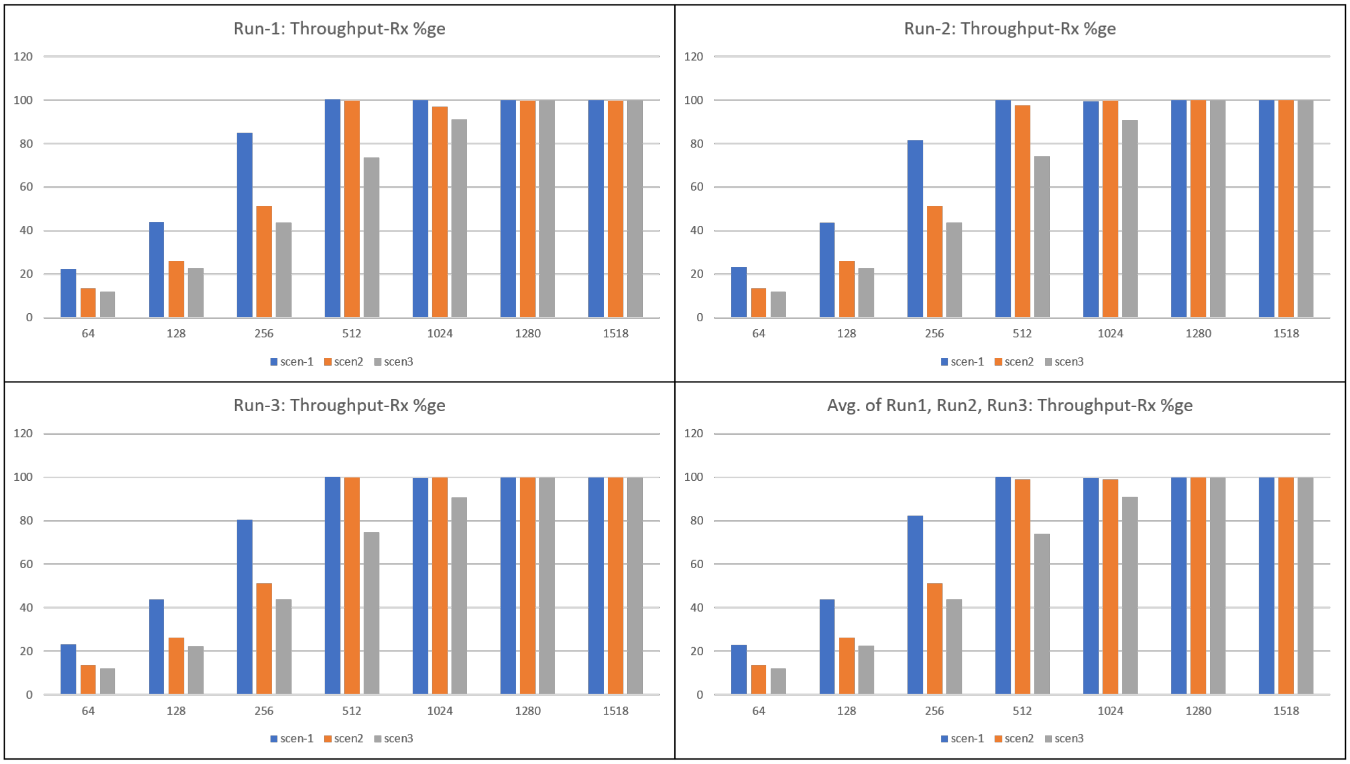
| | | | |
|---|---|---|---|
| 3 | PMDs: 4, 5, 6, 7<br>(0xF0) | VNF: 8, 22 | DUT: eno5, eno6<br>TGEN: 5, 6 |
| 4 | PMDs: 4,5,22,23<br>(0xC00030) | VNF: 8,9 | DUT: eno5, ens801f2<br>TGEN: 5, 7 |
| 5 | PMDs: 4,5, 22, 23<br>(0xC00030) | VNF: 24, 25 | DUT: eno5, ens801f2<br>TGEN: 5, 7 |
| 6 | PMDs: 4, 5, 22, 23<br>(0xC00030) | VNF: 8, 24 | DUT: eno5, ens801f2<br>TGEN: 5, 7 |
| 7 | PMDs: 22, 23, 24, 25<br>(0x3C00000) | VNF: 26, 27 | DUT: ens801f2, ens802f3<br>TGEN: 7, 8 |
| 8 | PMDs: 22, 23, 24, 24<br>(0x3C00000) | VNF: 4,5 | DUT: ens801f2, ens802f3<br>TGEN: 7, 8 |
| 9 | PMDs: 22, 23, 24, 25<br>(0x3C00000) | VNFs: 4,26 | DUT: ens801f2, ens802f3<br>TGEN: 7, 8 |

# Results: V2V

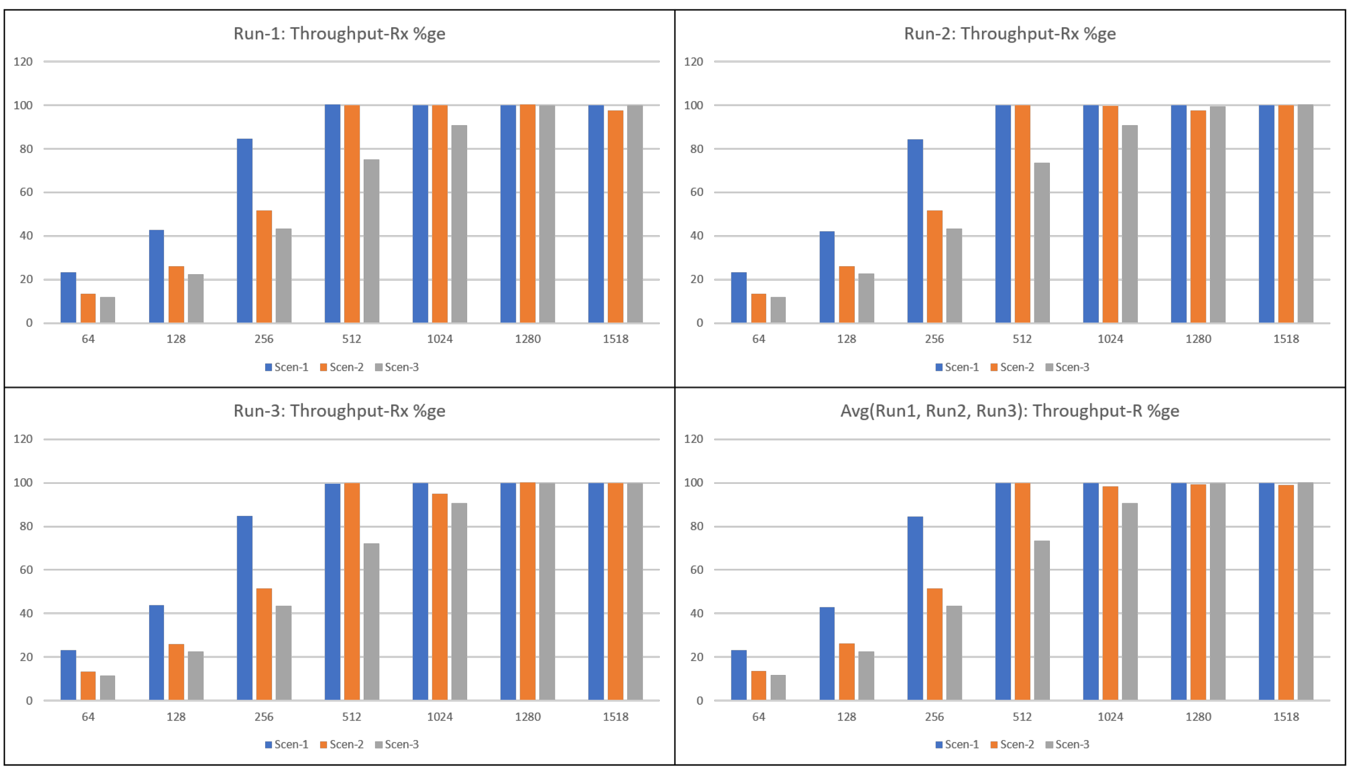
## RFC2544 Throughput Test Results




## RFC2544 With Loss Verification Throughput Test Results

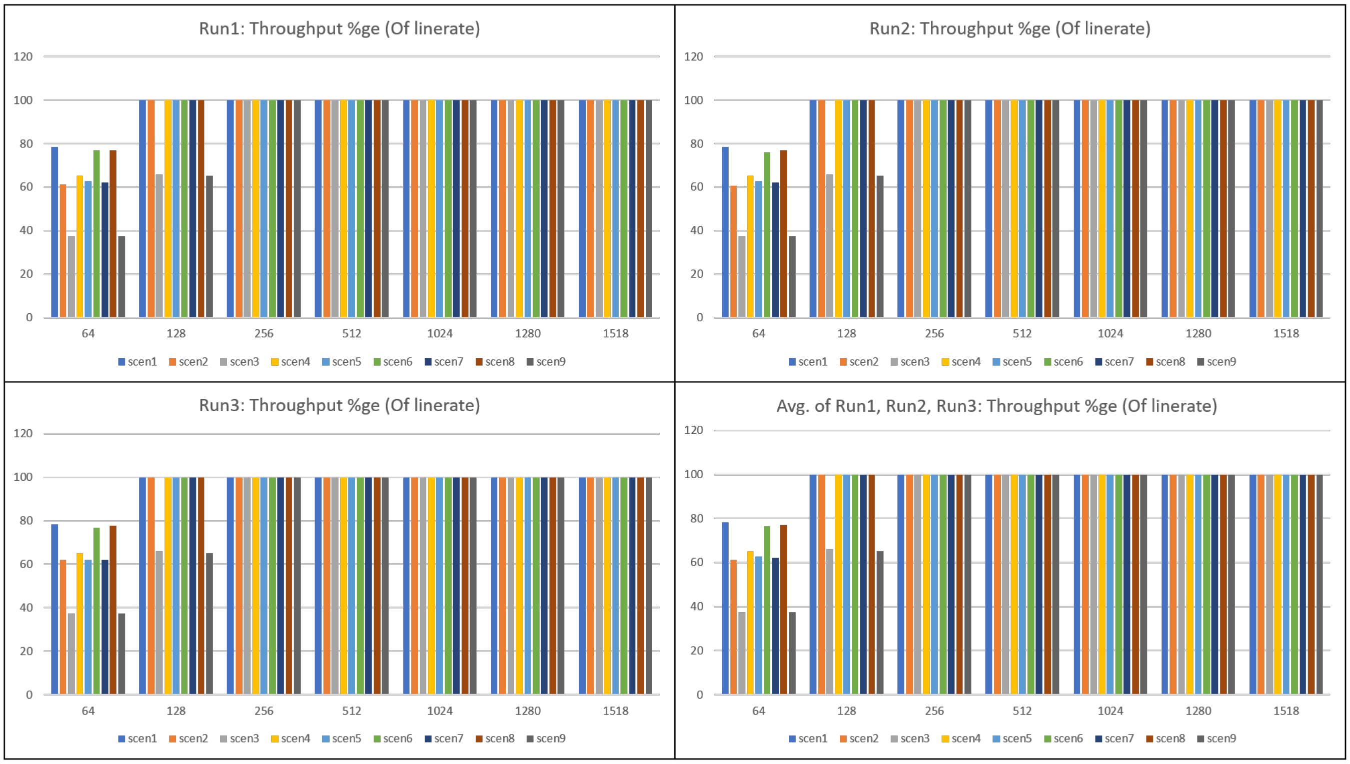



## Continuous Throughput Test Results

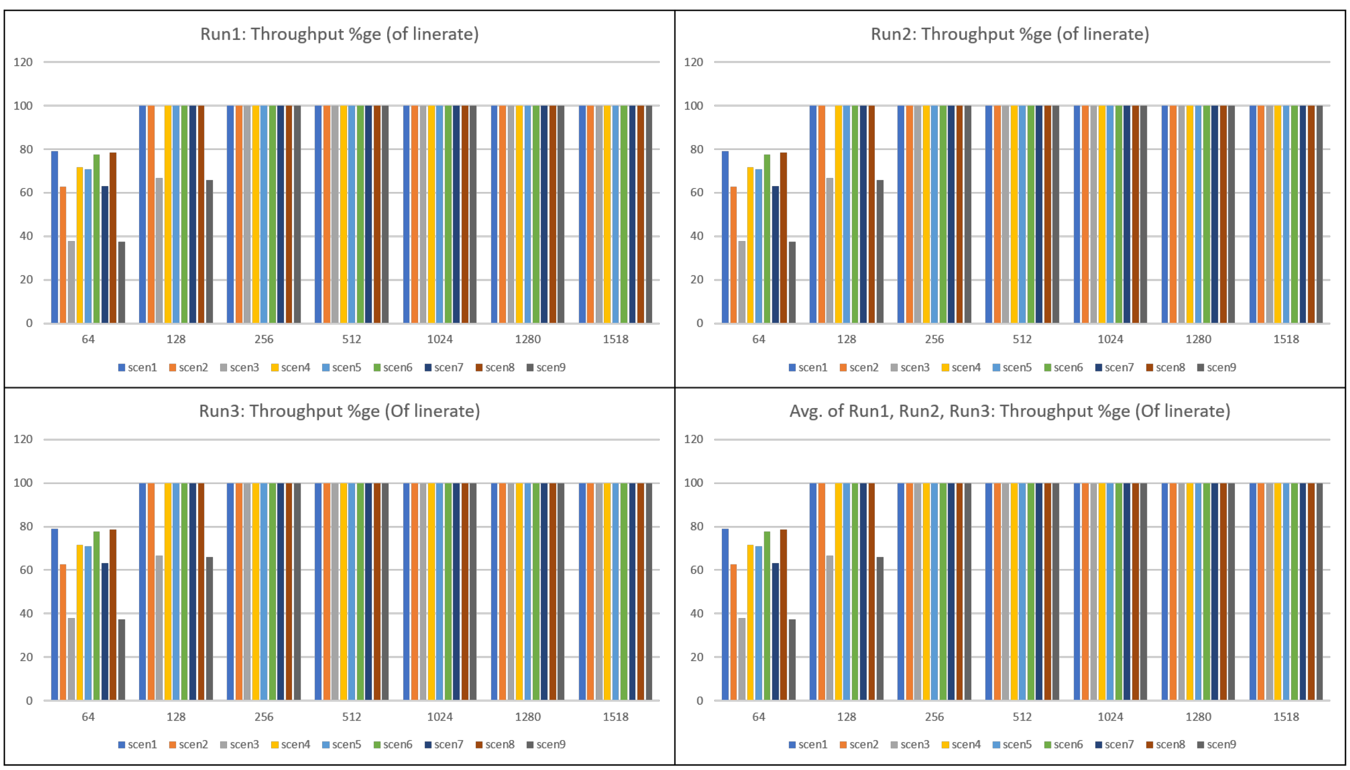



# Results: P2P

## RFC2544 Throughput Test Results

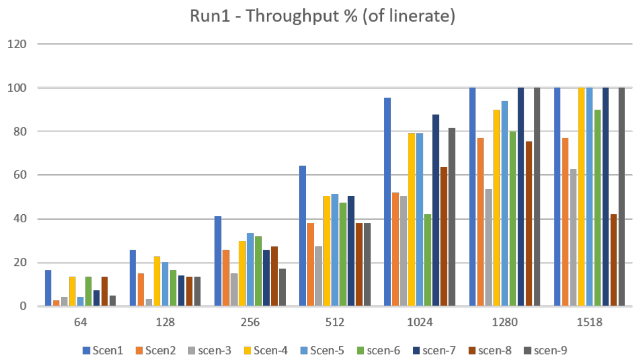


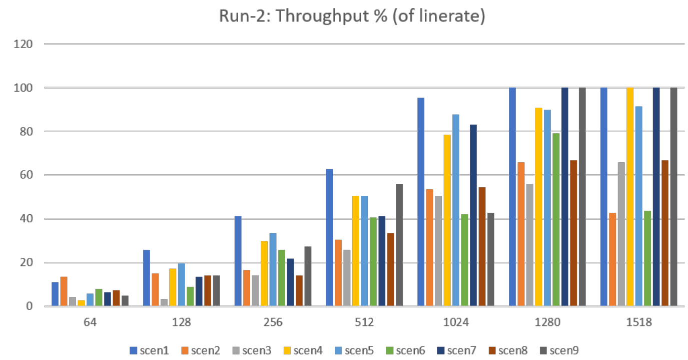

Continuous Throughput Test Results (Max Received Frame Rate at 100% of Line rate offered load)

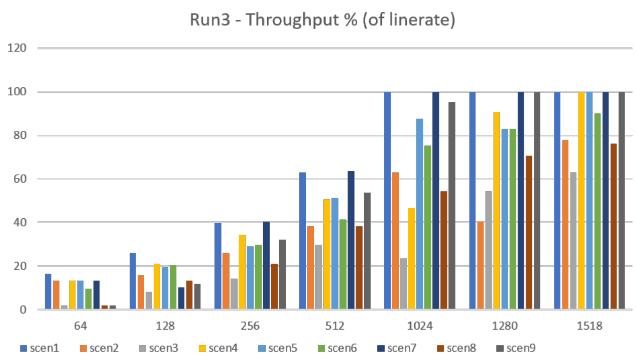


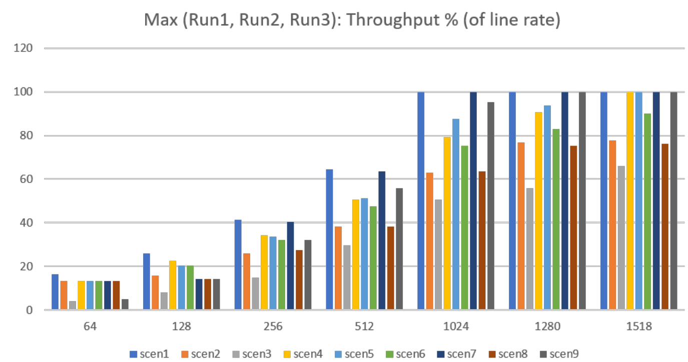

# Results: PVP

## RFC2544 Throughput Test Results

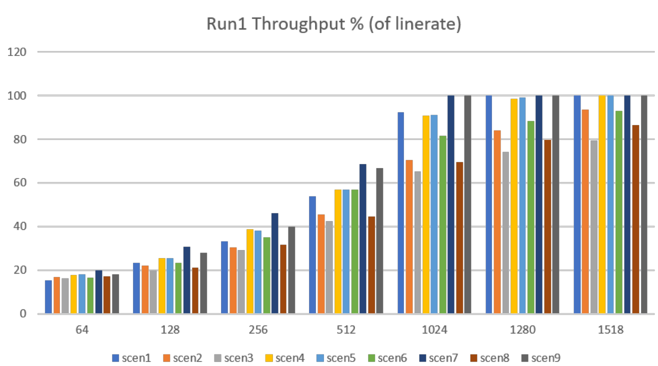


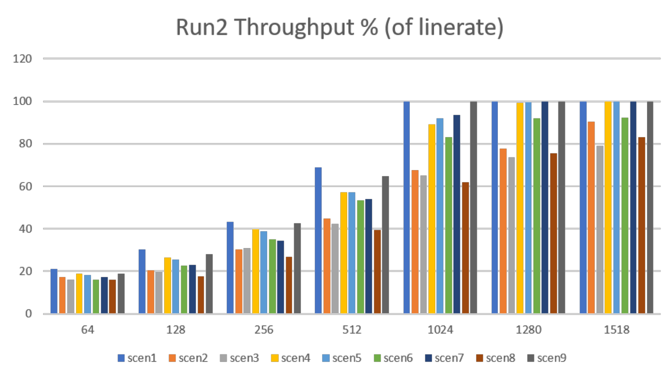



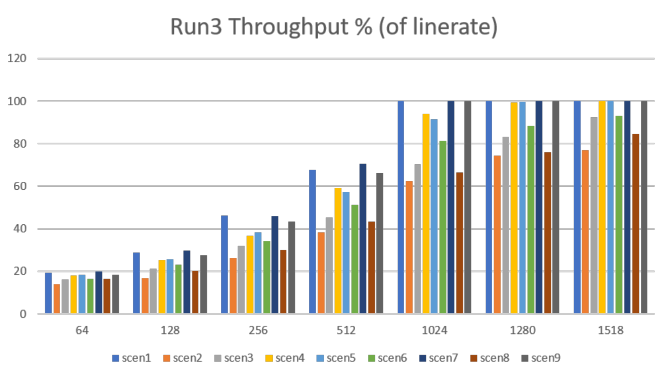



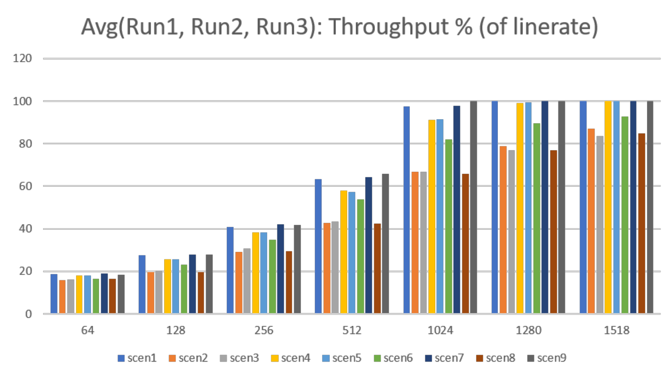




## Continuous Throughput Test Results (Max Received Frame Rate at 100% of Line rate offered load)

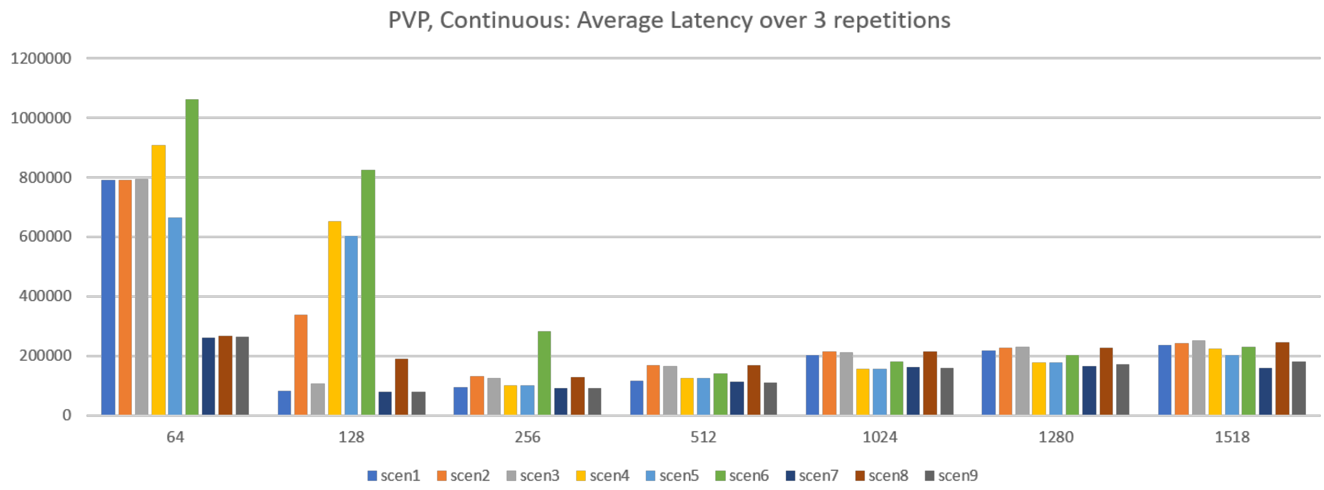


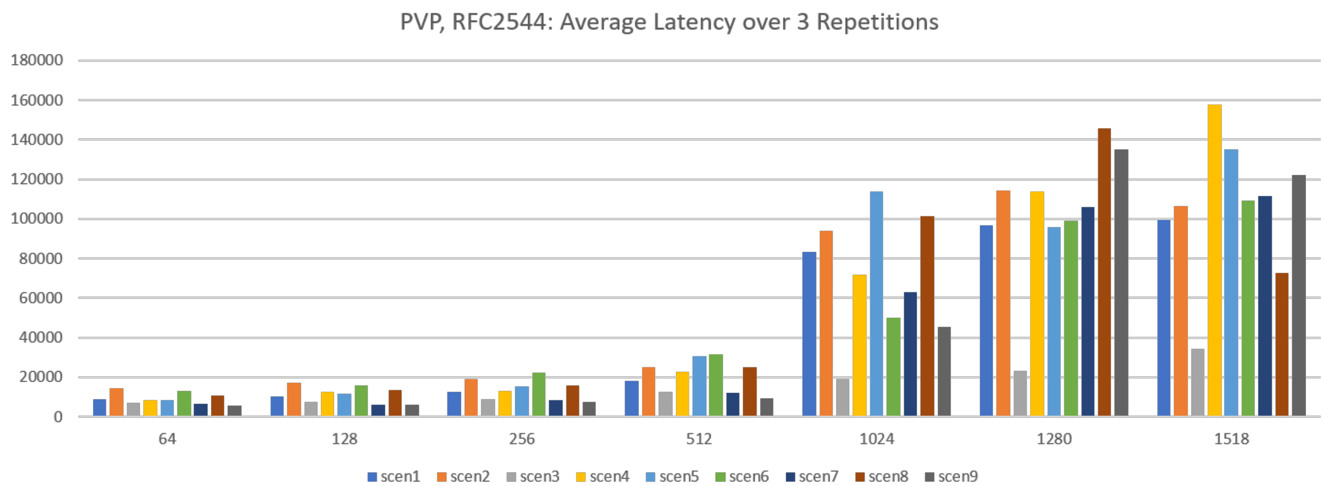




## PVP Latency Results




# Inferences

Theme: What is expected, What is unexpected,

## V2V:

1. Performance differences upto 1024 bytes packets sizes can be seen.
2. Single vCPU serving more interfaces is worse than CPU on the other numa serving the interfaces - This pattern is also seen in other (P2P, and PVP) scenarios.
3. RFC2544 with Loss-Verification is more consistent across runs, compared with RFC2544 without loss verification.

## P2P:

1. Only the smaller (64 and 128) packet sizes matter. For packets sizes above 128 the throughput performance remains similar.
2. Scenarios 2 and 7 can be seen as the worst case scenarios with both the PMD-cores running on different NUMA than the NIC. As expected, the performance is consistently low for both scenarios-2 and 7.
3. Interesting cases are Scenario-3 and Scenario-9.  Here a single pmd-core ends up serving both the NICs. This results in poorer performance than Scenario-2 and 7.
4. Scenario 1, 6, and 8 can be seen as good cases where each of the NICs are served by single, separate PMD-cores.
5. When one NIC is served by pmd-core on the same NUMA, whereas the other NIC is served by pmd-core on a different NUMA - Scenarios 4 and 5 - can be seen as average cases with lower performance than 1, 6 and 8 - but not as low as 3, 9, 2, and 7.
6. There is no difference in performance between continuous and RFC2544-throughput traffic tests.

## PVP:

Note: In these scenarios, we ensure there is always at least 1 PMD mapped to a NUMA to which a physical NIC is mapped to. That is, we will not encounter the case of Scenario-2 and 7 of the P2P here.

1. Continuous traffic results are more consistent across runs compared to RFC2544-throughput test.
2. The inconsistency across the runs in RFC2544 cases can be explained by the way the binary-search algorithm works - and, this can be used to argue about the importance of adaptive RFC2544 Binary-search algorithm in virtualized environments.
3. Due to cross-numa traffic flow, scenarios 2, 3 and 8, as expected, performs poorer compared to other scenarios.
4. When the NICs are mapped to both the NUMAs - with pmd-cores also present - the performance is similar across all movements of VNF cores. The scenarios 4, 5 and 6 represent these cases. However, among these, Scenario-6 is relatively poorer as its cores are split across NUMAs, and the chances are that only one of them would be used effectively.
5. Scenarios 1, 7 and 9 are the best cases - with minimal to none cross-numa effects.

## Generic:

1. X-NUMA instantiation is a very realistic scenario.  If we seek more realism, we might add a stressor load to a few of the interesting scenarios. This might enhance the effects of X-NUMA deloyment.

# Observations

## V2V Scenarios OVS_PMD and interfaces (virtual) mappings

| Scenarios | Mappings |
|---|---|
| Virtual Interfaces | **Bridge trex_br**<br>Port trex_br<br>Interface trex_br<br>type: internal<br>Port "dpdkvhostuser3"<br>Interface "dpdkvhostuser3"<br>type: dpdkvhostuser<br>Port "dpdkvhostuser2"<br>Interface "dpdkvhostuser2"<br>type: dpdkvhostuser<br>**Bridge "int_br0"**<br>Port "dpdkvhostuser0"<br>Interface "dpdkvhostuser0"<br>type: dpdkvhostuser<br>Port "dpdkvhostuser1"<br>Interface "dpdkvhostuser1"<br>type: dpdkvhostuser<br>Port "int_br0"<br>Interface "int_br0"<br>type: internal |
| Scenario-1 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdkvhostuser1 queue-id: 0<br>port: dpdkvhostuser2 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br>port: dpdkvhostuser0 queue-id: 0<br>port: dpdkvhostuser3 queue-id: 0 |
| Scenario-2 | **pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdkvhostuser0 queue-id: 0<br>port: dpdkvhostuser3 queue-id: 0<br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdkvhostuser1 queue-id: 0<br>port: dpdkvhostuser2 queue-id: 0 |

| | |
|---|---|
| Scenario-3 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdkvhostuser0 queue-id: 0<br>port: dpdkvhostuser1 queue-id: 0<br>port: dpdkvhostuser2 queue-id: 0<br>port: dpdkvhostuser3 queue-id: 0<br>**pmd thread numa_id 1 core_id 22:**<br>isolated : false |

## PVP Scenarios OVS-PMD and Interfaces (physical and virtual) mappings

| Scenario | Mappings |
|---|---|
| 1/2/3 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdkvhostuser1 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br>port: dpdk1 queue-id: 0<br>**pmd thread numa_id 0 core_id 6:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>**pmd thread numa_id 0 core_id 7:**<br>isolated : false<br>port: dpdkvhostuser0 queue-id: 0 |
| 4 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdkvhostuser1 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>port: dpdkvhostuser0 queue-id: 0<br>**pmd thread numa_id 1 core_id 22:**<br>isolated : false<br><br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdk1 queue-id: 0 |
| 5 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br><br>**pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdkvhostuser1 queue-id: 0<br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdk1 queue-id: 0<br>port: dpdkvhostuser0 queue-id: 0 |
| 6 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdkvhostuser1 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>port: dpdkvhostuser0 queue-id: 0<br>**pmd thread numa_id 1 core_id 22:**<br>isolated : false<br><br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdk1 queue-id: 0 |

| | |
|---|---|
| 7/8/9 | **pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdkvhostuser1 queue-id: 0<br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>**pmd thread numa_id 1 core_id 24:**<br>isolated : false<br>port: dpdkvhostuser0 queue-id: 0<br>**pmd thread numa_id 1 core_id 25:**<br>isolated : false<br>port: dpdk1 queue-id: 0 |

## P2P Scenarios OVS-PMDs and Physical-Interface Mappings

| Scenario | Mappings |
|---|---|
| 1 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdk1 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br>port: dpdk0 queue-id: 0 |
| 2 | **pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdk1 queue-id: 0 |
| 3 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>port: dpdk1 queue-id: 0<br>**pmd thread numa_id 1 core_id 22:**<br>isolated : false |
| 4 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdk1 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br>port: dpdk0 queue-id: 0 |
| 5 | **pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdk1 queue-id: 0 |
| 6 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>**pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdk1 queue-id: 0 |
| 7 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br>port: dpdk1 queue-id: 0<br>**pmd thread numa_id 0 core_id 5:**<br>isolated : false<br>port: dpdk0 queue-id: 0 |
| 8 | **pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>**pmd thread numa_id 1 core_id 23:**<br>isolated : false<br>port: dpdk1 queue-id: 0 |

| | |
|---|---|
| 9 | **pmd thread numa_id 0 core_id 4:**<br>isolated : false<br><br>**pmd thread numa_id 1 core_id 22:**<br>isolated : false<br>port: dpdk0 queue-id: 0<br>port: dpdk1 queue-id: 0 |

# Possible Variations

1. Increase the Number of CPUs to 4 for the VNF.
2. Phy2phy case (no VNF).
3. Try different forwarding VNF
4. Different Virtual Switch (VPP)
5. RxQ Affinity.

# Summary of Key Results and Points of Learning

1. Performance degradation due to Cross-NUMA Node instantiation of NIC, vSwitch, and VNF can vary from 50-60% for lower size packets (64, 128, 256) to under 0-20% for higher packet sizes ( > 256 bytes)
2. The worst performance was observed with PVP setup and scenarios where all PMD cores and NICs are in same NUMA Node, but VNF cores are shared across NUMA Nodes. Hence, VNF cores are best allocated within the same NUMA Node. If the VIM prevents VNF instantiation across multiple NUMA Nodes then, this issue is effectively avoided. However, at the time of this testing, K8s is believed to be NUMA Node agnostic when placing Pods in its normal mode, or when using DPDK and SR/IOV, and makes this testing more relevant until the situation changes.
3. Any variations in CPU assignments under P2P setups has no effect on performance for packet sizes above 128 bytes. However, V2V setups show performance differences for larger packet sizes of 512 and 1024 bytes.
4. Continuous traffic-tests and RFC2544 Throughput using Binary Search with Loss-Verification provides more consistent results across multiple-runs. Hence, these methods should be preferred over legacy RFC2544 methods using Binary search or Linear search algorithms.
5. A single NUMA Node serving multiple interfaces is worse than Cross-NUMA Node performance degradation. Hence, it is better to avoid such configurations. For example, if both the physical NICs are assigned to NUMA-Node id 0 (with core ids 0-21), then the configuration-a below will lead to poorer performance than configuration-b

**Configuration-a**

pmd thread **numa_id 0** core_id 4:

isolated : false

port: dpdk0 queue-id: 0

port: dpdk1 queue-id: 0

pmd thread **numa_id 1** core_id 22:

isolated : false

**Configuration-b**

pmd thread **numa_id 1** core_id 22:

isolated : false

port: dpdk0 queue-id: 0

pmd thread **numa_id 1** core_id 23:

isolated : false

port: dpdk1 queue-id: 0

6. The average latencies have exactly opposite patterns under PVP setups and scenarios for continuous traffic testing and RFC2544 throughput test (with search algorithm BSwLV). That is, average latency is lower for lower packet sizes for RFC2544 throughput test and higher for higher packet-sizes, and this trend is opposite for continuous traffic testing.

Note: For result 6, could this be the result of the continuous traffic testing filling all queues for the duration of the trial? The RFC 2544 Throughput methods (and those of the present document) allow the queues to empty and the DUT to stabilize between trials.

Notes on Documentation

1. must view log files, qemu threads need to match the intended scenario for VM -
2. Christian created qemu command (and documentation) - check this for VM mapping
3. SR: CT's command is only the host
4. qemu command line **-smp 2** should do this - simulates two Numa Nodes - need to see how the VM see it's architecture: **numactl -h**