Benchmarking Virtual Switches in OPNFV
draft-vsperf-bmwg-vswitch-opnfv-01

- Maryam Tahhan
  Al Morton

Outline

- VSPERF test specification updates
- VSPERF in practice
- Future work
- Summary

VSPERF test specification updates

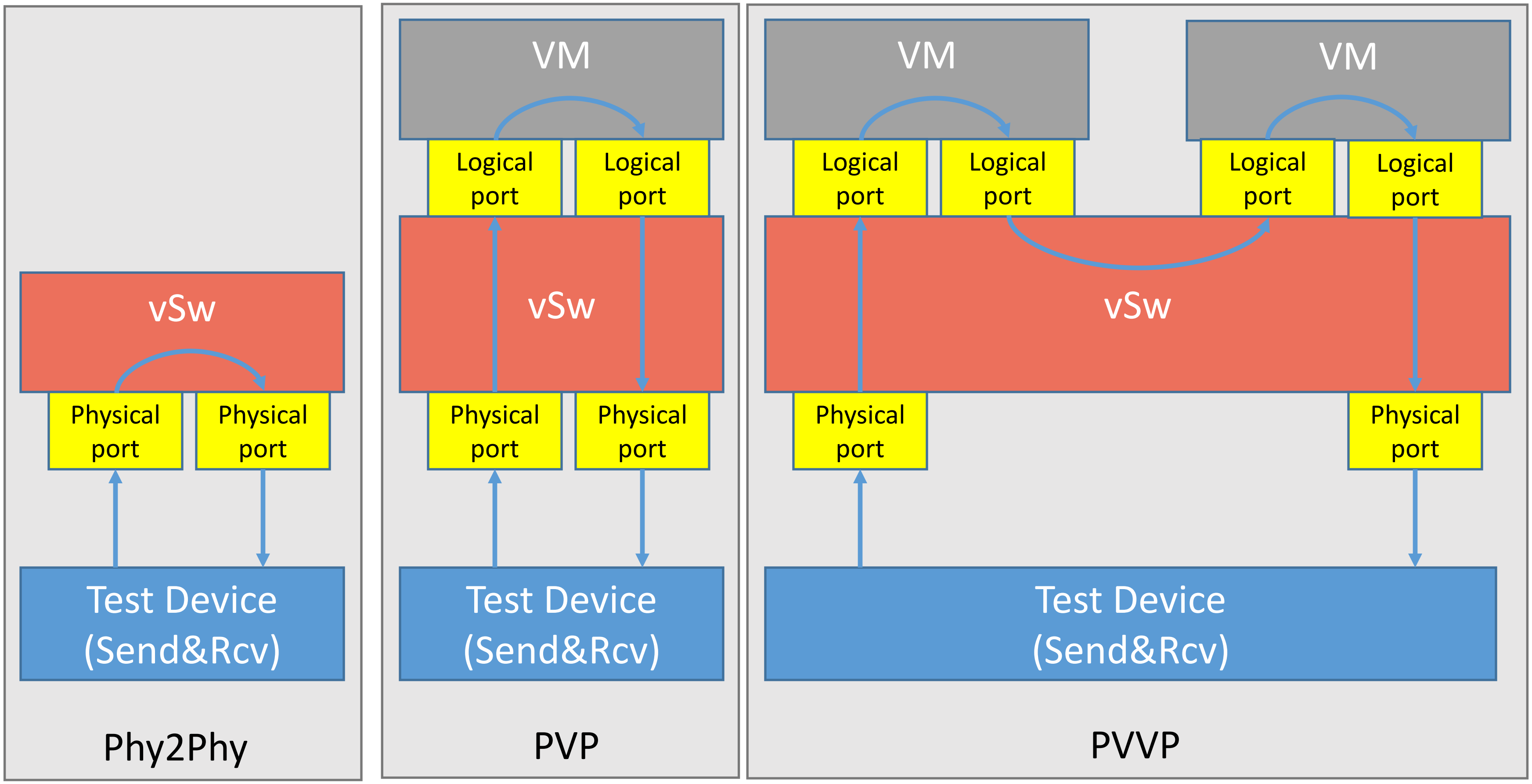
- New tests have been added to extend the matrix coverage.
- Soak tests were migrated from using “RFC2544 Throughput” to referring to “RFC 2889 Maximum Forwarding Rate”.
- Refined the Fully-Meshed RFC 2889 tests to include deployment and to report the number of ports used for the test.
- Scalability tests now look at the situations where flows are not installed and pre installed on the switch.

Matrix Coverage of the Current LTD

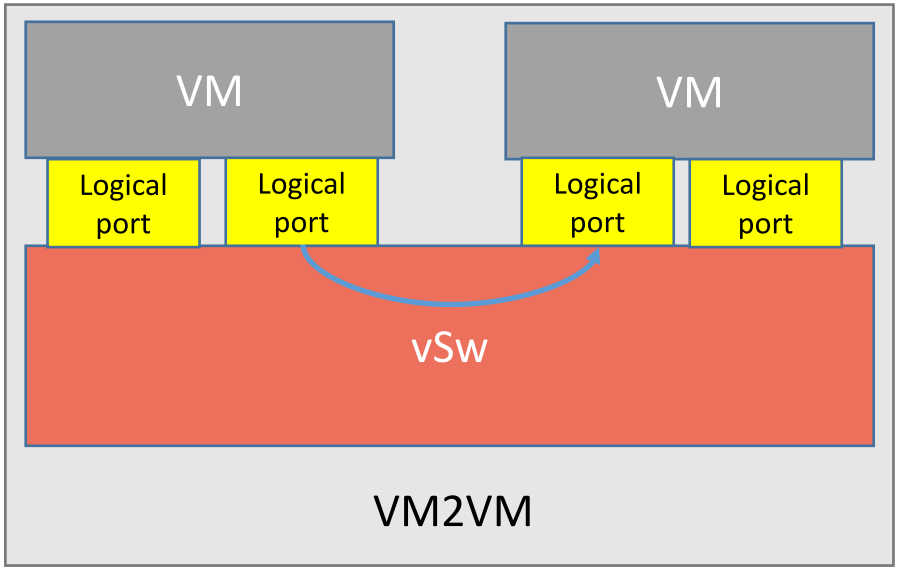
	SPEED	ACCURACY	RELIABILITY	SCALE
Activation	• Activation.RFC2889.AddressLearningRate • Activation.RFC2889.AddressCachingCapacity • PacketLatency.InitialPacketProcessingLatency	• CPDP.Coupling.Flow.Addition	• Throughput.RFC2544.SystemRecoveryTime • Throughput.RFC2544.ResetTime	• Throughput.RFC2889.AddressCachingCapacity
Operation	• Throughput.RFC2544.PacketLossRatio • Throughput.RFC2544.PacketLossRateFrmMod • Throughput.RFC2544.BackToBackFrames • Throughput.RFC2889.MaxForwardingRate • Throughput.RFC2889.ForwardPressure • Throughput.RFC2889.BroadcastFrameForwarding • RFC2889 Broadcast Frame Latency test • CPU.RFC2544.0PacketLoss	• Throughput.RFC2889.ErrorFramesFiltering • Throughput.RFC2544.Profile	• ~~Throughput.RFC2544.Soak~~ → Throughput.RFC2889.Soak • ~~Throughput.RFC2544.SoakFrameModification~~ → Throughput.RFC2889.SoakFrameModification • PacketDelayVariation.RFC3393.Soak	• Scalability.RFC2544.0PacketLoss • MemoryBandwidth.RFC2544.0PacketLoss.Scalability
De-Activation				

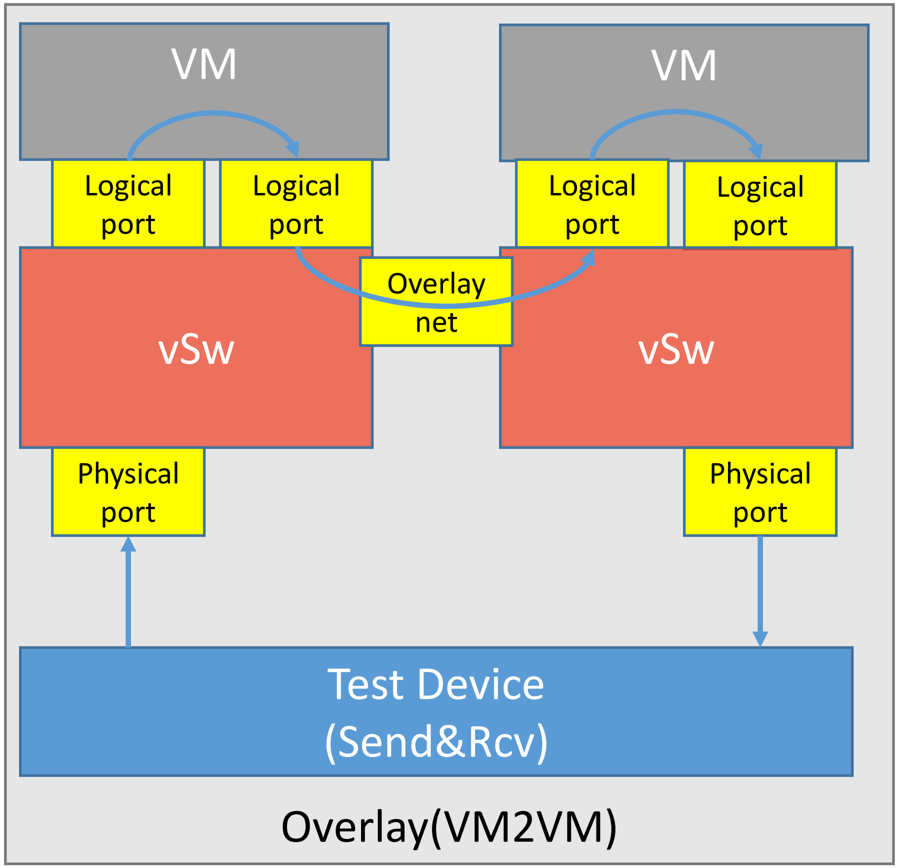
New tests in white.

VSPERF LTD Supported Deployment Scenarios



VSPERF LTD Supported Deployment Scenarios cont.

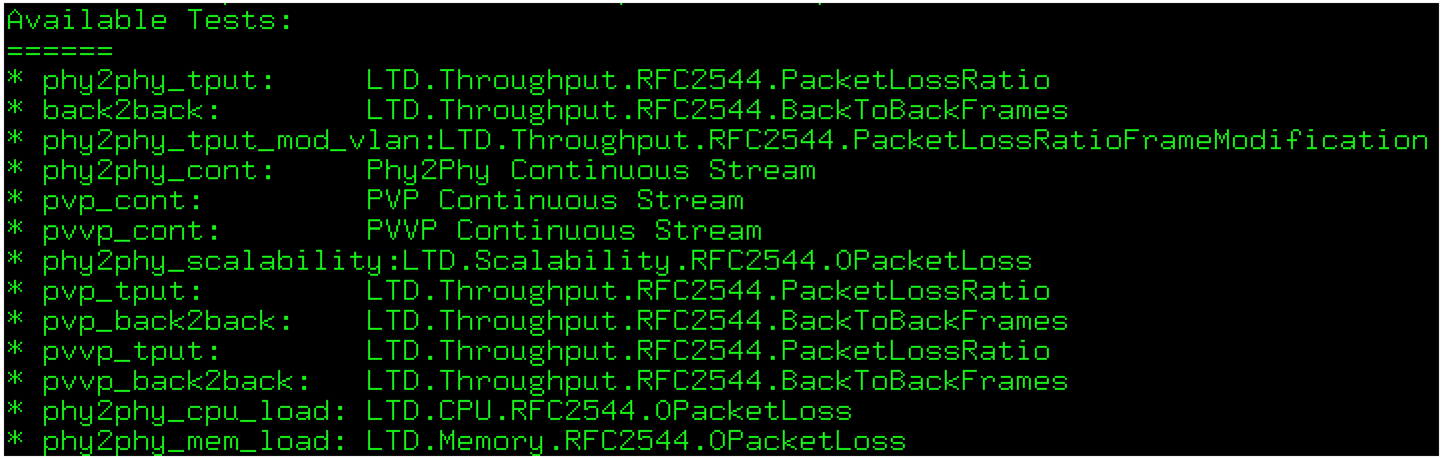




In Practice

VSPERF Test Framework

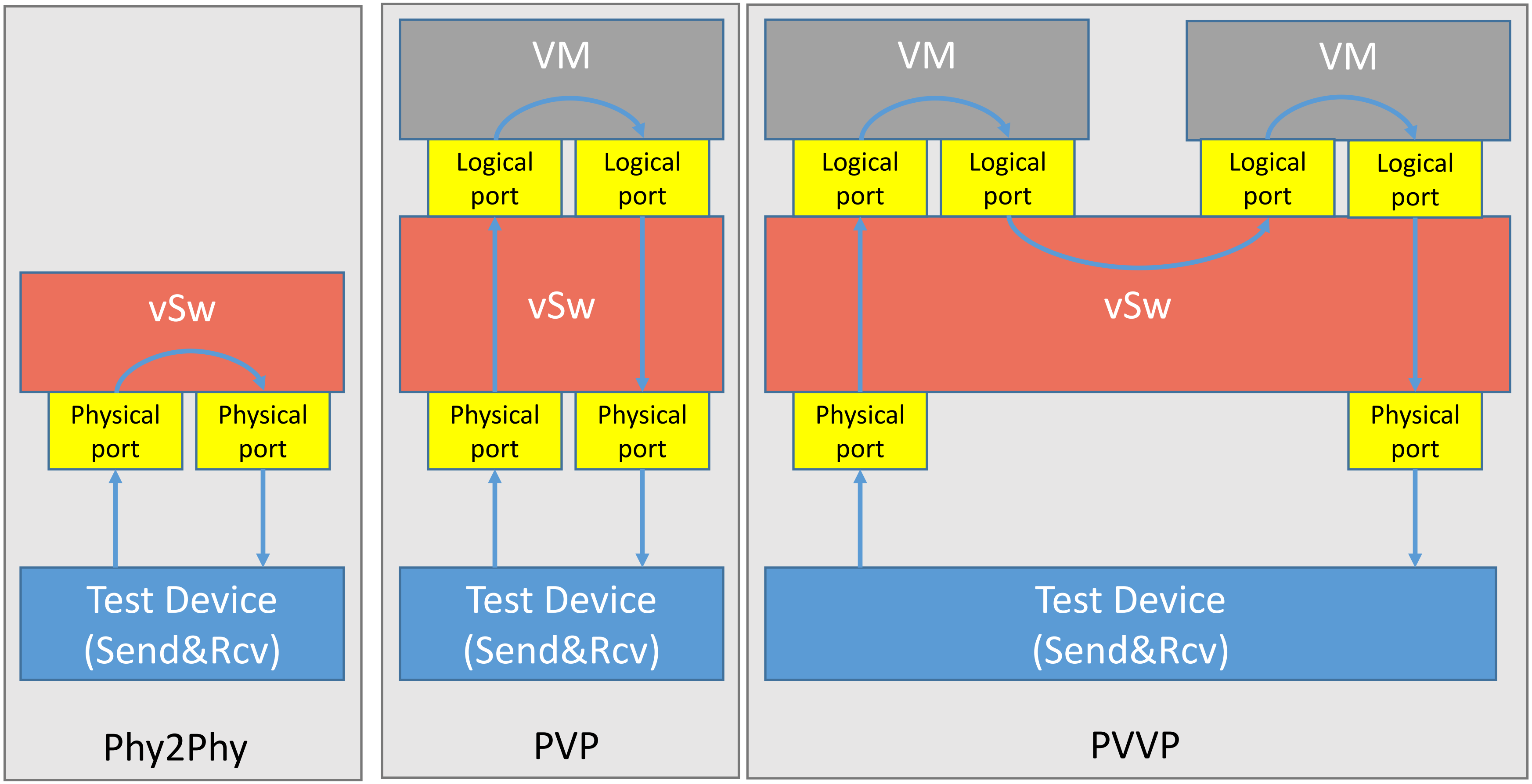
- A Python based test framework for characterizing the performance of virtual switches.
- Used to prove out and refine the tests and the methodologies for VSPERF.
- As of today, capable of conducting the following tests on stock OVS and OVS with DPDK:



```
Available Tests:
======
* phy2phy_tput:     LTD.Throughput.RFC2544.PacketLossRatio
* back2back:        LTD.Throughput.RFC2544.BackToBackFrames
* phy2phy_tput_mod_vlan:LTD.Throughput.RFC2544.PacketLossRatioFrameModification
* phy2phy_cont:     Phy2Phy Continuous Stream
* pvp_cont:         PVP Continuous Stream
* pvvp_cont:        PVVP Continuous Stream
* phy2phy_scalability:LTD.Scalability.RFC2544.0PacketLoss
* pvp_tput:         LTD.Throughput.RFC2544.PacketLossRatio
* pvp_back2back:    LTD.Throughput.RFC2544.BackToBackFrames
* pvvp_tput:        LTD.Throughput.RFC2544.PacketLossRatio
* pvvp_back2back:   LTD.Throughput.RFC2544.BackToBackFrames
* phy2phy_cpu_load: LTD.CPU.RFC2544.0PacketLoss
* phy2phy_mem_load: LTD.Memory.RFC2544.0PacketLoss
```

- Supported deployment scenarios to date: Phy2Phy, PVP and PVVP.

VSPERF Framework Supported Deployment Scenarios



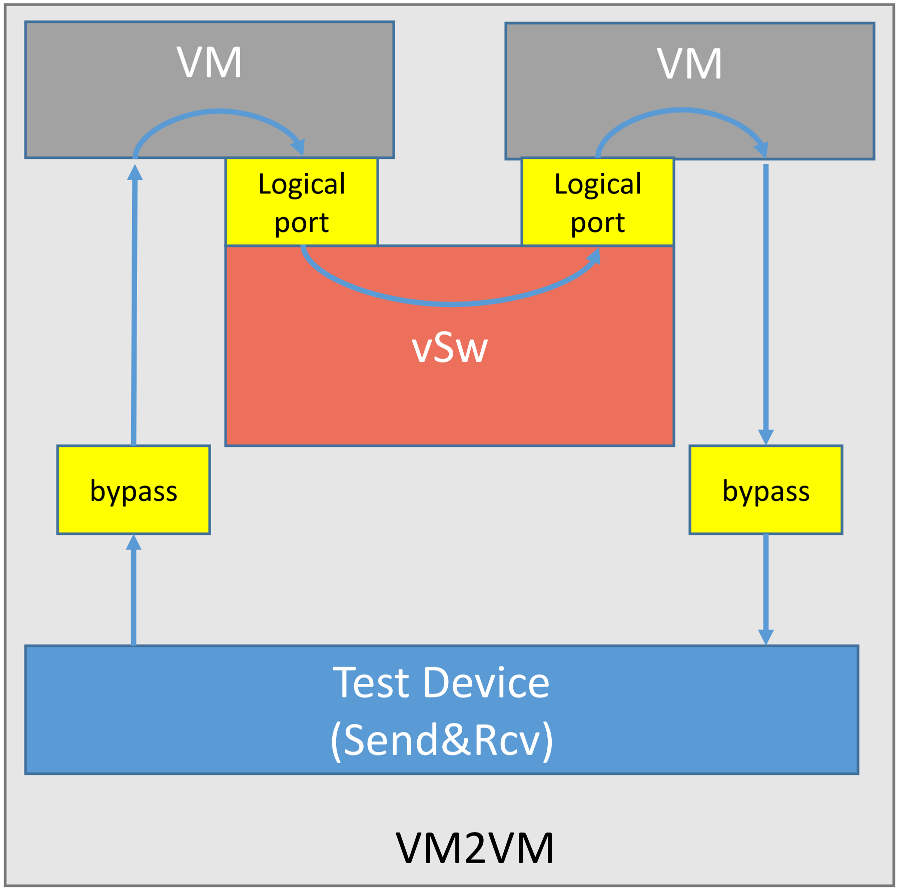
Matrix Coverage of the Current Test Framework

	SPEED	ACCURACY	RELIABILITY	SCALE
Activation	* Activation.RFC2889. AddressLearningRate * Activation.RFC2889. AddressCachingCapacity *PacketLatency.Initial PacketProcessingLatency	*CPDP.Coupling.Flow.Addition	* Throughput.RFC2544. SystemRecoveryTime * Throughput.RFC2544. ResetTime	* Throughput.RFC2889. AddressCachingCapacity
Operation	* Throughput.RFC2544.PacketLossRatio * Throughput.RFC2544. PacketLossRateFrmMod * Throughput.RFC2544. BackToBackFrames * Throughput.RFC2889. MaxForwardingRate * Throughput.RFC2889. ForwardPressure * Throughput.RFC2889. BroadcastFrameForwarding * RFC2889 Broadcast Frame Latency test * CPU.RFC2544.0PacketLoss	* Throughput.RFC2889. ErrorFramesFiltering * Throughput.RFC2544.Profile	* ~~Throughput.RFC2544. Soak~~ → Throughput.RFC2889.Soak * ~~Throughput.RFC2544. SoakFrameModification~~ → Throughput.RFC2889.SoakFrameModification * PacketDelayVariation.RFC3393.Soak	* Scalability.RFC2544.0PacketLoss * MemoryBandwidth.RFC2544.0PacketLoss.Scalability
De-Activation				

Implemented tests in white for Phy2Phy, PVP and PVVP.

VM2VM in Practice

- Hasn’t been implemented yet
- Concerns around time synchronization between VMs and clock accuracy.
- Recommendation under consideration: Test must include an external HW traffic generator to act as the tester/traffic source and sink.



Future Work

- Integrating multiple traffic gens: Spirent, Moongen and Xena. (current IXIA)
- Methodology extensions: Iterations for the short trial tests
- Prove out and refine methodology and tests through the framework
- Add more tests to the LTD and the framework, an initial list:
    - Scalability Tests adding More VMs in succession and building a performance profile as we add more VMs.
    - Overlay Networking Tests: VXLAN performance testing, encap, decap, encap and decap.
    - Match action performance testing? The cost of the different actions supported by a vSwitch.
    - Classifying L2, L3 and L4 traffic Profile Tests.
    - Stream/bulk Data transfer "unidirectional stream" performance.
    - Request & response/transaction rate tests.
    - Performance testing with Mirroring enabled on the switch.
    - TCP Max connections per second, Max # of active sessions, Max transactions per second.
    - IPv6 considerations
    - Best of N and Worst of N Tests
    - Deactivation tests

Summary

- The LTD and the test framework will be developed continuously for some time.
- We would like your opinion on:
  - **WG Adoption** of this Summary Draft as a snapshot of next OPNFV Release (Brahmaputra),
    - with pointers to Released and current versions of LTD spec and VSPERF as it grows/evolves.
    - Eventually, Convert entire LTD spec to an Internet Draft/RFC
  - Whether we should continue to provide periodic updates on the expanding/evolving LTD Spec.

BACKUP

What is OPNFV?

Open Platform for NFV Project (OPNFV):

- A Linux Foundation open source project focused on accelerating the evolution of Network Functions Virtualization (NFV).
- OPNFV will establish a **carrier-grade, integrated, open source reference platform** for NFV that ensures consistency, performance and interoperability among multiple open source components.
- OPNFV will work with upstream projects to coordinate continuous integration and testing while filling development gaps.

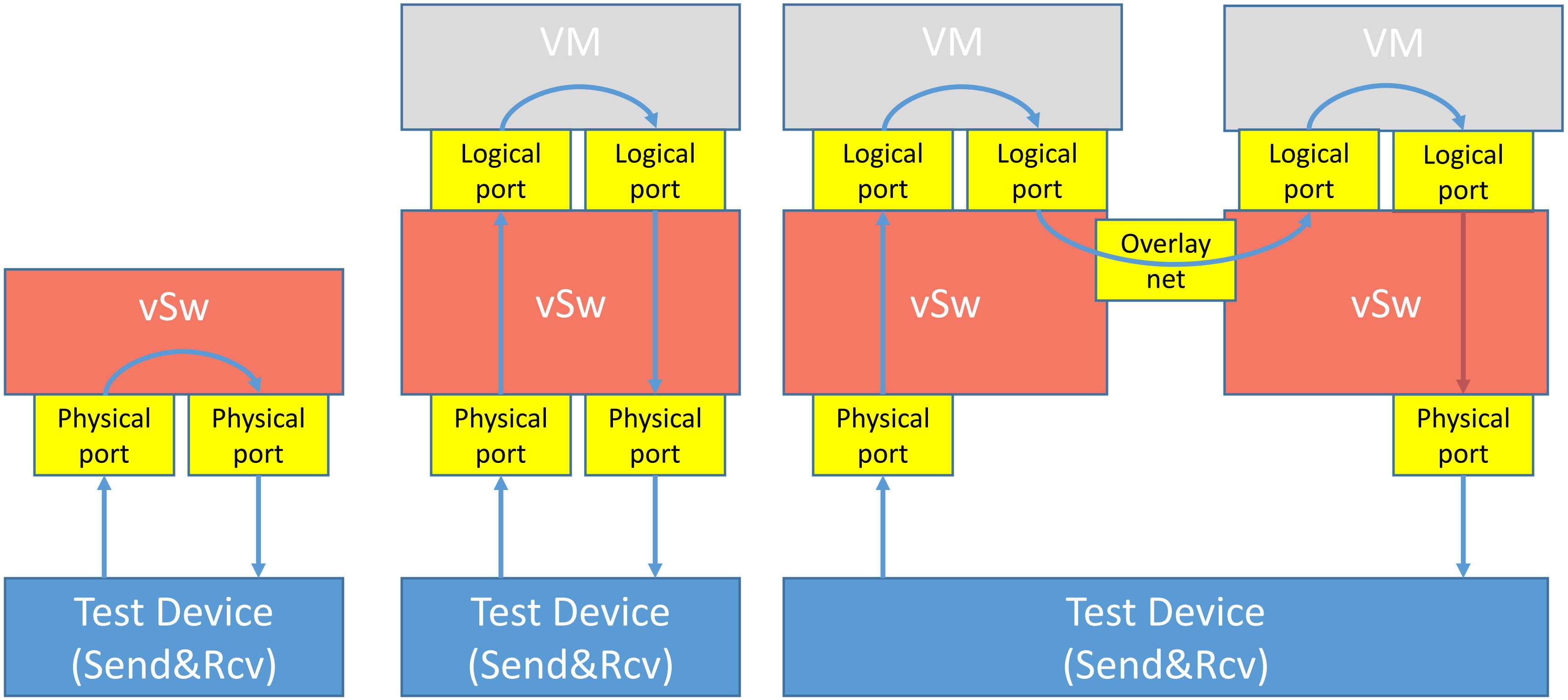
How can I join OPNFV?

- Create a Linux Foundation account that you will use for all the tools provided by the Linux Foundation. You also need this account to contribute to OPNFV projects.
- To participate, via contribution, in any project in OPNFV, you will need to contact the project manager/lead for the project.
- Project Roles: contributor, committer, and project lead.

What is VSWITCHPERF AKA VSPERF?

- An OPNFV Project
- Goal: Characterize the performance of a virtual switch for **Telco** NFV use cases.
- Virtual switches have not typically been designed for Telco NFV use cases that require Telco grade determinism in their performance and support for latency/jitter-sensitive Telco traffic.
- This project proposes defining and executing an appropriate set of tests in order to objectively measure the current Telco characteristics of a virtual switch in the NFVI

Additional Test Setups (single traffic direction shown)



vSwitch deployment scenarios

- Physical port → vSwitch → physical port .
- Physical port → vSwitch → VM → vSwitch → physical port .
- Physical port → vSwitch → VM → vSwitch → VM → vSwitch → physical port .
- Physical port → vSwitch → VM.
- VM → vSwitch → physical port.
- VM → vSwitch → VM.

Please note a Physical port is connected to a traffic generator. A VM is connected to the vSwitch through a logical port.