



Making Networking Apps Scream on Windows with DPDK

Jason Messer, Microsoft, Principal PM Lead

Manasi Deval, Intel, Sr. Staff Engineer

DPDK Summit - San Jose – 2017



#DPDKSummit

Legal Notices and Disclaimers



- ▶ Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.
- ▶ No computer system can be absolutely secure.
- ▶ Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit <http://www.intel.com/performance>.
- ▶ Intel, the Intel logo, Xeon and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.
- ▶ © 2017 Intel Corporation.

Agenda

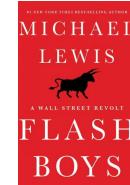


- ▶ Motivation for Fast Packet Processing on Windows
- ▶ The journey to bring DPDK to Windows
- ▶ What happened to PacketDirect?
- ▶ Roadmap
- ▶ Q&A

Motivation – Time is Money



Fractions of a second (latency) can make all the difference in Olympic races, financial gain or loss in the stock market, and winners and losers in gaming competitions



- ▶ Windows developers need a low-latency, high-throughput network data path for
 - ▶ Gaming and Video Streaming
 - ▶ Cloud Infrastructure
 - ▶ Unified Communications
 - ▶ IoT Edge Gateways
 - ▶ Network Intrusion Detection/Prevention
 - ▶ Other mobile operator network appliances

The Journey: High-Performance Networking on Windows



- ▶ Native Host (Kernel-Mediated IO) Software and Hardware Offloads
- ▶ Virtualized Host Offloads with Hyper-V Virtual Switch
- ▶ Guest VFs with SR-IOV
 - ▶ Multi-tenancy support - Watch this space...
- ▶ User-Mode Applications with RIO Sockets (~750k packets per second)
- ▶ DPDK on Windows...

The Journey: Bringing DPDK to Windows

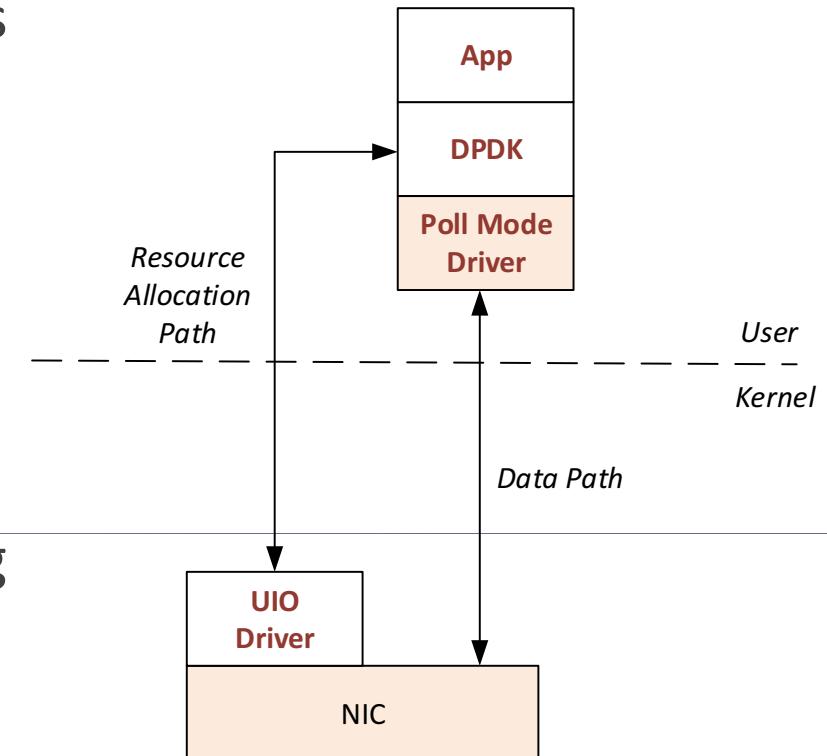


- ▶ Project Goals
 - ▶ Native DPDK Solution on Windows
 - ▶ Full parity (features, performance, et. al.) with Linux DPDK implementation
 - ▶ Easily enable other IHVs to support DPDK natively on Windows (Common Interfaces)
 - ▶ Release work to community as an Open-Source project

DPDK on Windows – Architecture

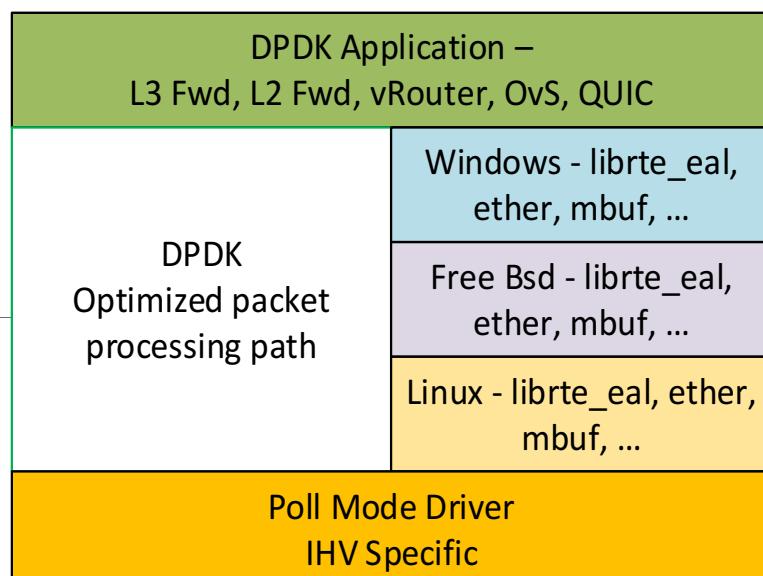


- DPDK libraries and application compiled under Windows
- Using Intel C/C++ compiler (ICC) for performance optimization
- UIO driver used to create physically contiguous memory block and provide user-mode mapping to HW resources
- Post-initialization, the Poll-mode driver (PMD) will send/receive packets directly to/from the NIC, bypassing all packet processing in kernel



Open Source

DPDK on Windows – Current Status



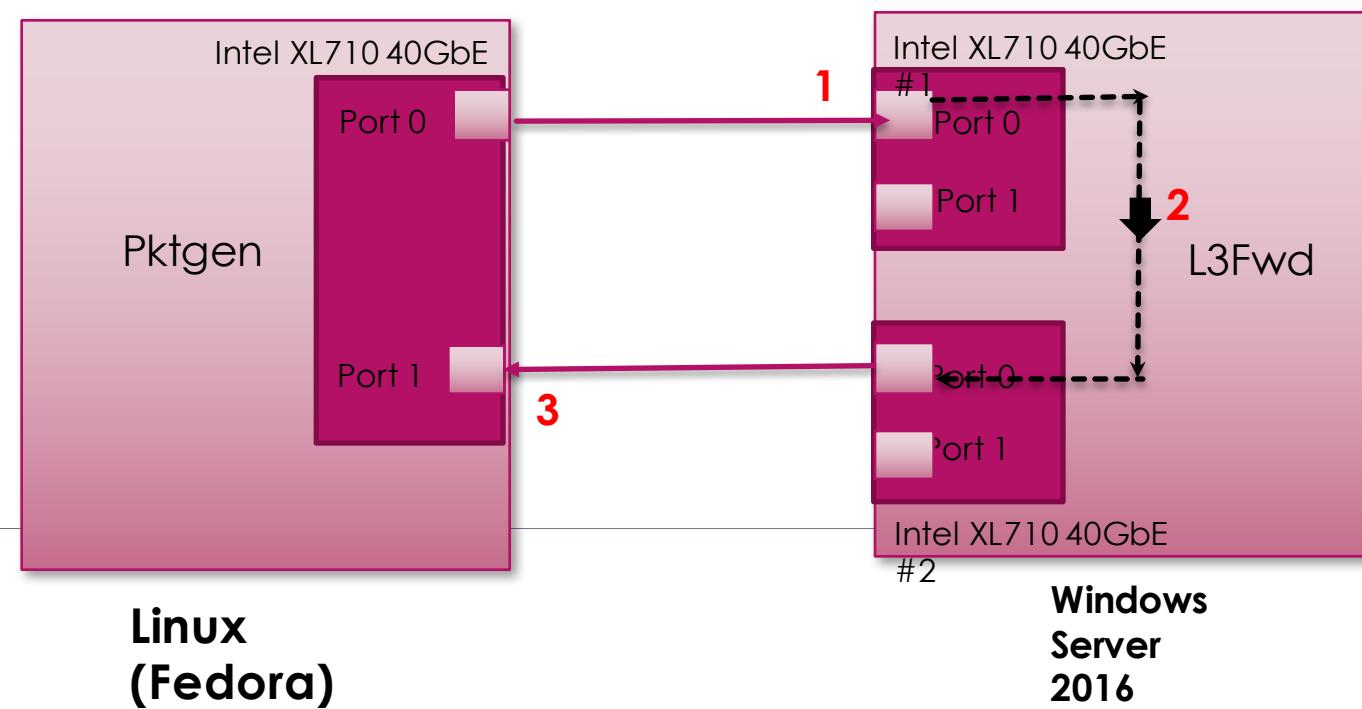
- Windows Server 2016 and Intel 40GbE network devices
- All essential DPDK libraries compiled and working in Windows
 - *librte_eal; librte_ether; librte_ring; librte_mempool; librte_mbuf* etc.
 - Update the i40e PMD to interface with the OS specific interface
- Challenge to create Windows OS compatible headers without using `#ifdef WINDOWS` in core DPDK
 - Use the “magic” of include file dependencies to allow Windows-related changes to be compiled into core DPDK code
 - Need work-arounds for a few GCC-specific implementations:
 - `typeof(x)` – not available in MS/Intel C compiler; requires an inelegant work-around
 - `__attribute__((constructor, used))` – needs an initialization work-around



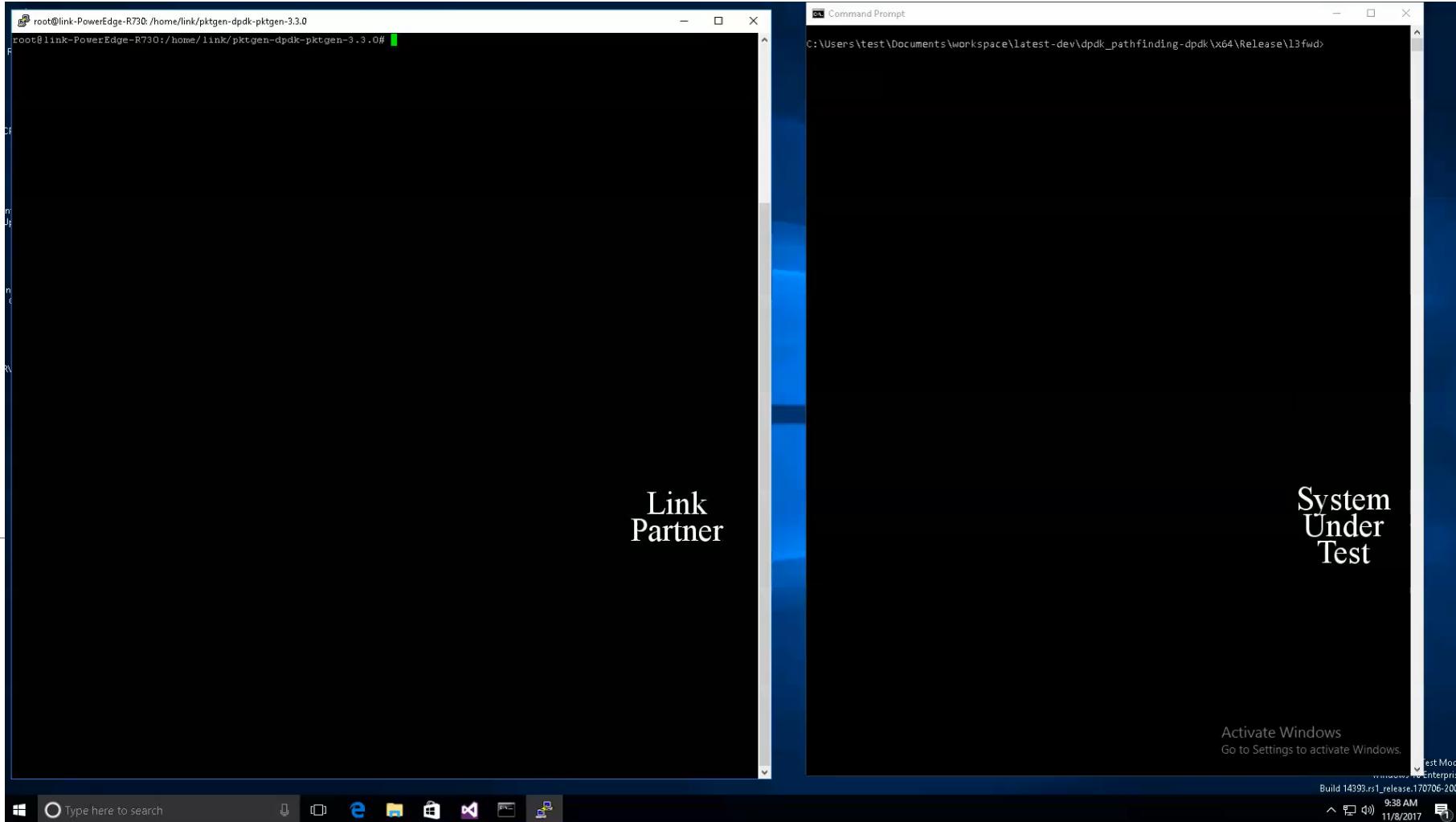
Demo

Manasi Deval, Intel

Demo Setup



Demo Video



Tested L2 FWD and L3 FWD applications.

Bi-directional Performance (using L3Fwd ↔ Pktgen)

Test system configuration

- Intel® Broadwell class 2U server
- Intel® Xeon® E5-2699 v4 CPU x 2
- 128GB RAM
- Intel® Ethernet Converged Network Adapter XL 710 (40GbE)
- Windows Server 2016

Packet Size (bytes)	Max pkts/sec (Mpps)
64	71.25665
128	67.01615
256	36.31707

DPDK Performance on Windows at par with Linux!

Code Availability



- ▶ UIO Driver and Microsoft Patches for DPDK v17.08
- ▶ Intending to mainline and publish soon... ☺



- ▶ Initial discussions and POCs starting at Microsoft
 - ▶ Network Security
 - ▶ Infrastructure Workloads
- ▶ Certain applications do not require a protocol stack such as TCP/IP
- ▶ New applications and microservices may include custom protocol stacks which are optimized for the specific workload

What about Kernel-Mode?



- ▶ PacketDirect (PD) - “experimental” feature available in Windows Server 2016 for Hyper-V Virtual Switch to accelerate path into and out of VM
- ▶ Learnings from PD around extensibility, feature gaps, and diagnosability resulted in an evolution to more inclusive design to address both server and client needs
- ▶ Design focus for user-mode vs kernel-mode applications
 - ▶ Accelerating User-Mode Network IO: DPDK
 - ▶ Accelerating Kernel-Mediated Network IO: Evolved PackedDirect
 - ▶ New driver model based on Windows Driver Framework (WDF)
 - ▶ New network data-path for Windows (WinSock APIs sit on top of this model)
 - ▶ More details coming soon... (Crawl, Walk, Run)

Microsoft loves Open Source Software (OSS)



- ▶ Over the past few years Microsoft transformed into a company that embraces open source
 - ▶ Docker
 - ▶ Kubernetes
 - ▶ Linux
- ▶ Towards this end, we have been working closely with Intel in bringing this DPDK submission to the open source community
- ▶ We want to develop an eco-system of high-performance applications, built on Windows, and would love to hear from you!

Roadmap: DPDK on Windows



- ▶ Upstream DPDK Code patches for Windows and contribute Windows UIO Driver to open source project
- ▶ Work with NIC Partners (IHVs) to bring Poll-Mode Drivers to Windows
- ▶ Ensure all DPDK libraries and APIs are fully functional on Windows (Close any gaps)
- ▶ Enable Co-Existence with other network stacks (e.g. for simple host management)
- ▶ Enable DPDK over IOV path into VM / Container with Guest VF
- ▶ [Longer Term] Protocol Stacks integration
- ▶ [Longer Term] Improve Security for multi-tenancy
- ▶ [Longer Term] Hardware Spreading and Steering

Questions?



Thank You



- ▶ **Call to Action:** Download, Build, and Run apps with DPDK on Windows!

- ▶ Jason Messer (jmesser@microsoft.com) @jmesser81 on Slack, GitHub, etc.
- ▶ Manasi Deval (deval.manasi@intel.com)
- ▶ Maintainer: Omar Cardona (ocardona@microsoft.com)

- ▶ Additional thanks to Intel contributors John (Mingtong) Lee, Ranjit Menon, Elizabeth Kappler, Miles Penner, Ashish Adhav, and Pallavi Kadam
- ▶ Thanks to Microsoft consultants Brandon Jiang, Omar Cardona, Jeff Tippett