DPDK

cryptodev

Overview, status & future work

Declan Doherty

DPDK Summit Userspace – Dublin, Oct 2016

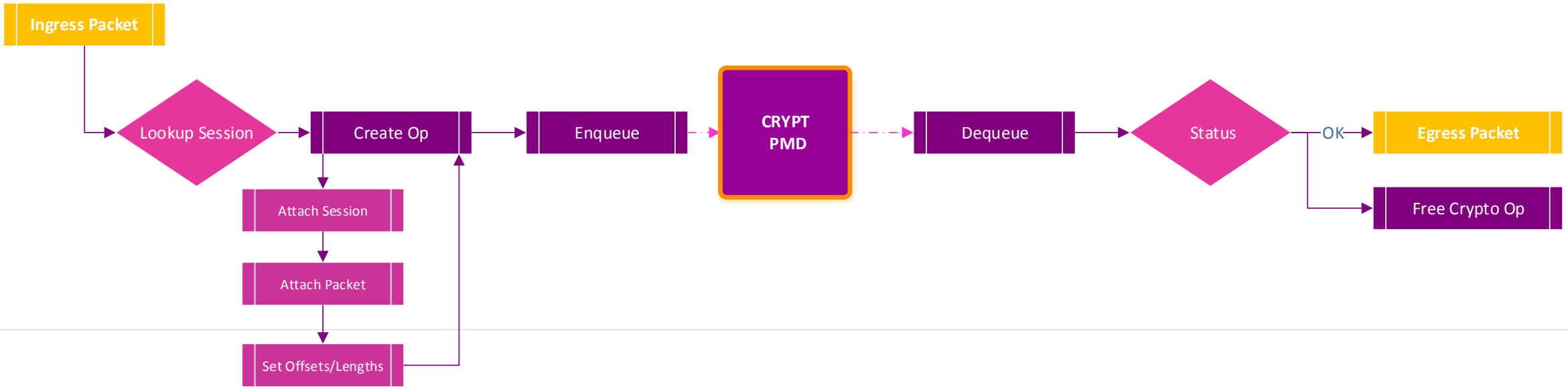
cryptodev

Overview

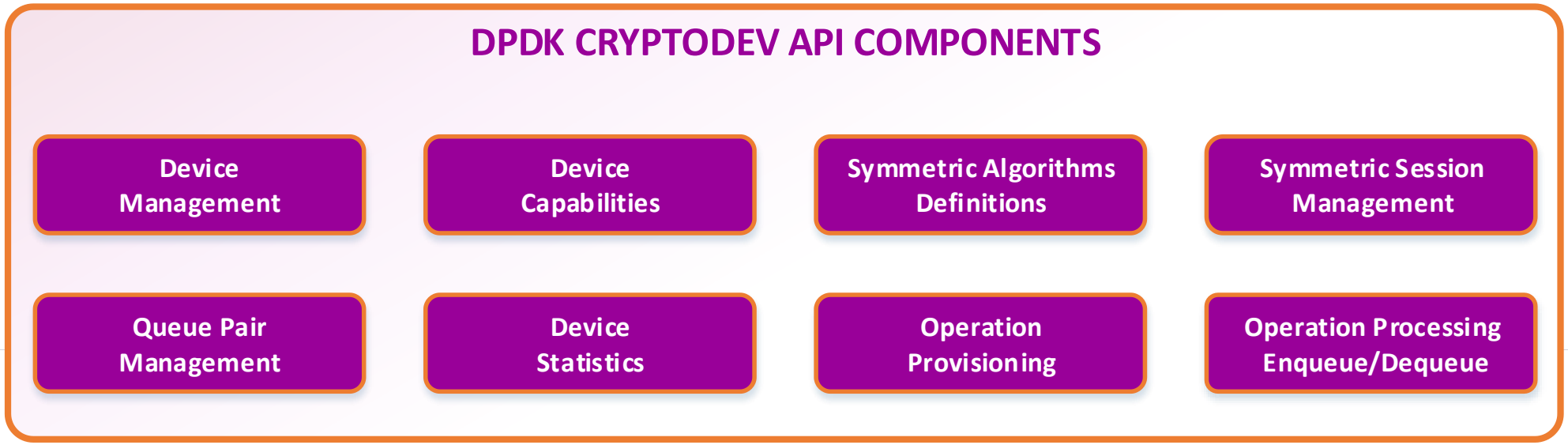
Overview

- Framework for processing symmetric crypto workloads in DPDK.
- Defines a standard API which supports both hardware accelerated lookaside and software based crypto processing.
- Underlying method of crypto operation processing is transparent to user application, allowing migration of work from hardware to software dynamically.
- Poll mode driver infrastructure for crypto devices.
- Supports cipher, authentication and AEAD symmetric crypto operations.
- Supports provisioning of chained cipher / authentication operations.
- Provide session management APIs
- Asynchronous burst processing API to amortise the cost of crypto operations across multiple packets and also to maximise performance when offloading to hardware accelerators.

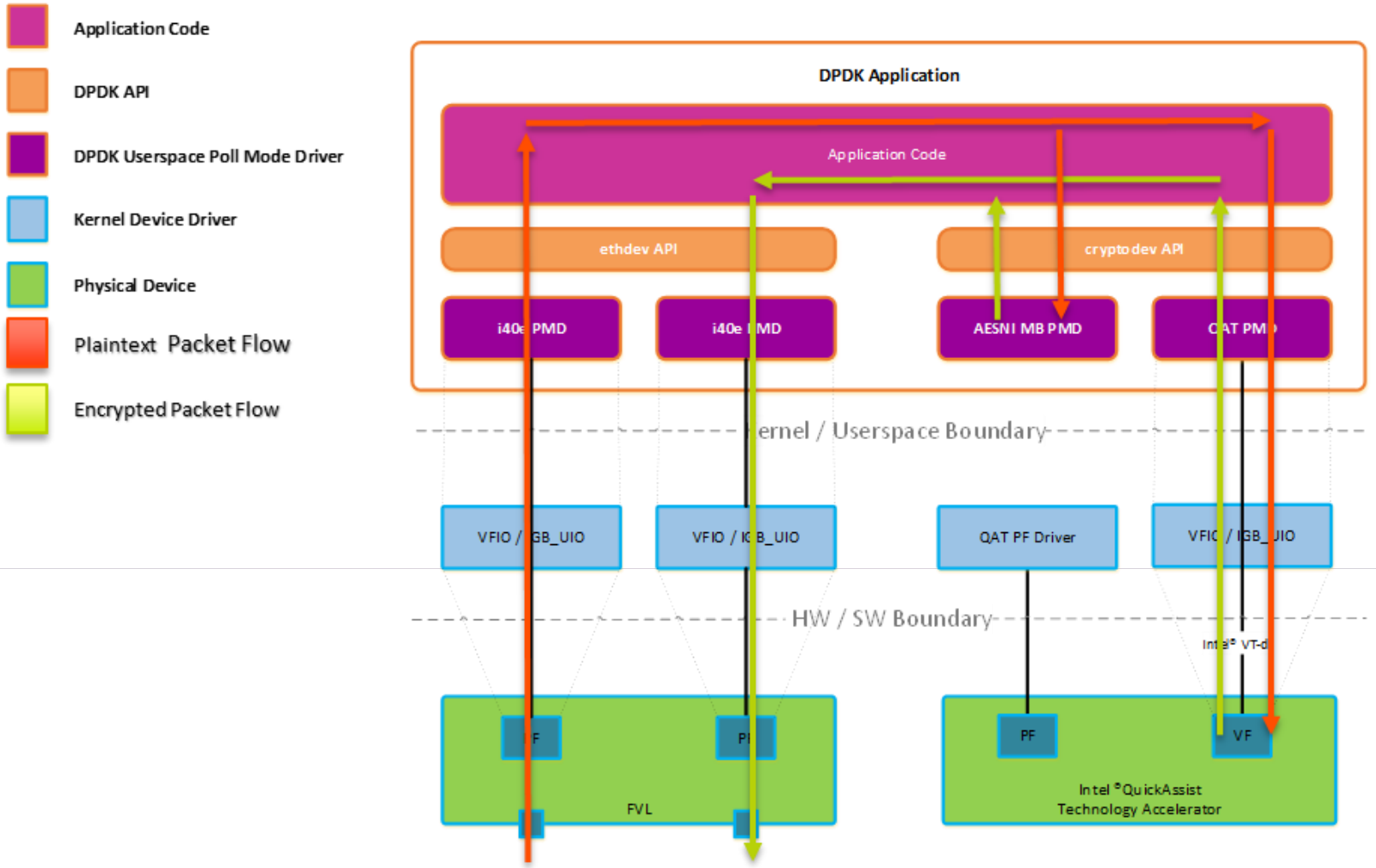
Crypto processing pipeline



API Components



Application Crypto Packet Processing Flow



cryptodev

Development

Feature delivery since userspace '15

R2.2
- Cryptodev Experimental Release
- AESNI MB PMD, QAT PMD

R16.04
- Cryptodev Stable Release
- Added AESNI GCM PMD, NULL PMD, Snow3G PMD
- Algorithm enablement QAT PMD

R16.07
- Kasumi PMD
- QAT PMD algorithm enablement

R16.11
- Kasumi PMD
- Openssl PMD
- QAT PMD algorithm enablement

* QAT = Intel(R) QuickAssist Technology

Changes from experimental release

- Moved from a rte_mbuf oriented

uint16_t **rte_cryptodev_enqueue_burst** (dev_id, qp_id, struct **rte_mbuf** **pkts, int16_t nb_pkts);

uint16_t **rte_cryptodev_dequeue_burst** (dev_id, qp_id, struct **rte_mbuf** **pkts, int16_t nb_pkts);

- to rte_crypto_op burst API

uint16_t **rte_cryptodev_enqueue_burst** (dev_id, qp_id, struct **rte_crypto_op** **ops, int16_t nb_ops);

uint16_t **rte_cryptodev_dequeue_burst** (dev_id, qp_id, struct **rte_crypto_op** **ops, int16_t nb_ops);

- Simplified mbuf management, no need for extra metadata on mbuf. No requirement to check if crypto_op needs to be freed on freeing of mbuf.
- Also stopped elements of cryptodev code being introduced into other parts of DPDK library code.

Crypto Poll Mode Drivers

- **crypto_aesni_gcm** – AESNI / vectorised accelerated software PMD
- **crypto_aesni_mb** – AESNI / vectorised accelerated software PMD
- **crypto_kasumi** – Vectorised accelerated software PMD
- **crypto_openssl** – PMD which shims crypto operations into the Openssl's libcrypto
- **crypto_null** – software PMD
- **crypto_qat** – Intel® QuickAssist Technology hardware accelerator
- **crypto_snow3g** – Vectorised accelerated software PMD
- **crypto_zuc** – Vectorised accelerated software PMD

Supported Algorithms

Algorithm	QAT*	SW VECTORIZED	SW LEGACY
AES GMAC 128-bit	✓		✓
AES GMAC 192-bit	✓		✓
AES GMAC 256-bit	✓		✓
AES XCBC 128-bit	✓	✓	
KASUMI F9		✓	
MD5			✓
MD5_HMAC	✓	✓	✓
SHA1			✓
SHA1_HMAC	✓	✓	✓
SHA224			✓
SHA224_HMAC	✓	✓	✓
SHA256			✓
SHA256_HMAC	✓	✓	✓
SHA384			✓
SHA384_HMAC	✓	✓	✓
SHA512			✓
SHA512_HMAC	✓	✓	✓
SNOW3G UIA2	✓	✓	
ZUC EEA3		✓	

Algorithm	QAT*	SW VECTORIZED	SW LEGACY
3DES CBC 128-bit	✓		✓
3DES CBC 192-bit	✓		✓
3DES CTR 128-bit			✓
3DES CTR 192-bit		✓	✓
AES CBC 128-bit	✓	✓	✓
AES CBC 192-bit	✓	✓	✓
AES CBC 256-bit	✓	✓	✓
AES CTR 128-bit	✓	✓	✓
AES CTR 192-bit	✓	✓	✓
AES CTR 256-bit	✓	✓	✓
KASUMI F8		✓	
NULL	✓	✓	✓
SNOW3G UEA2	✓	✓	
ZUC EEA3		✓	

Algorithm	QAT	SW VECTORIZED	SW LEGACY
AES GCM 128-bit	✓	✓	
AES GCM 192-bit	✓		
AES GCM 256-bit	✓		

* QAT = Intel(R) QuickAssist Technology

cryptodev

Future Work

Framework / PMDs

- Adding rte_mbuf scatter-gather support to all software crypto PMDs
- Migration of crypto_aesni_gcm to ISA-L crypto, enabling AES-GCM 256bit
- Cipher only/ authentication only operations to crypto_aesni_mb
- PCI Hot-plug support to framework
- Crypto operation performance optimisations

Crypto Performance Application

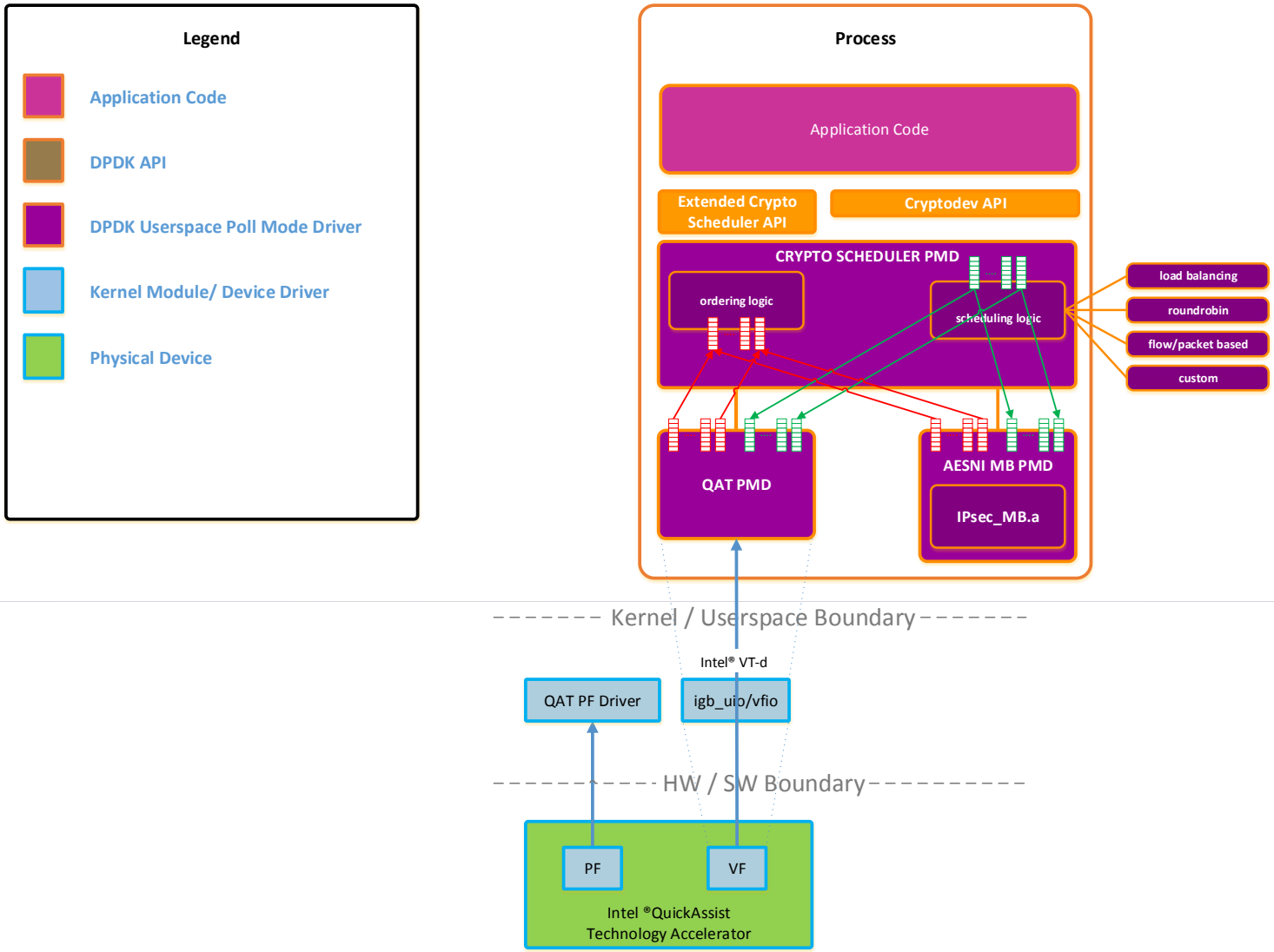
- New application to enable benchmarking of crypto PMD performance on any system.
- Modular to allow any crypto PMD to be tested if it can support the algorithm combination.
- Allows configuration of all components of the PMD and all configuration of elements of the crypto operations to be executed.
- Will support throughput and latency measurement initially.

```
./crypto-perf $eal_options -- --ptest throughput --devtype crypto_aesni_gcm --optype aead --cipher-algo aes-gcm --cipher-op encrypt --cipher-key-sz 16 --auth-algo aes-gcm --auth-op generate --auth-key-sz 16 --auth-aad-sz 12 --auth-digest-sz 16 --total-ops 10000000 --silent --burst-sz 32 --buffer-sz 2048
```

Crypto Scheduler

- Allows multiple crypto devices to be slaved under a single device.
- Investigating per queue, per flow and per packet scheduling paradigms.
- Pluggable scheduling/ordering logic, may allow user to define there own scheduler and dynamically load.
- Many scheduling modes being investigated:
  - fat flow – load balancing a single flow across multiple hardware accelerators
  - sw fallback – allow flows to processed on core when hw accelerator is oversubscribed.
  - per packet scheduling – packet size / session type and PMD utilization used to decide crypto PMD to use.
  - distributor – balance across many cores for sw crypto

Crypto Scheduler PMD



Questions?

Declan Doherty

declan.doherty@intel.com

Enabling IPSec Cryptodev Offload

Sergio González Monroy

Network Software Engineer @ Intel

DPDK Summit Userspace - Dublin- 2016

Agenda

- IPSec Development
- Enabling Cryptodev in FD.IO/VPP
- Preliminary Performance
- Future Work
- Questions

IPSec Development

DPDK 16.04

- IPSec-secgw sample app
- Basic data path functionality
- AES-CBC
- HMAC-SHA1-96
- ESP tunnel

DPDK 16.07

- Transport Mode
- IPv6 support

DPDK 16.11

- AES-GCM
- AES-CTR
- Config file

VPP 16.06

- OpenSSL libcrypto
- IKEv2 (responder only)
- Anti-Replay Window
- Extended Sequence Number (ESN)
- Nested SAs
- IPSec interface (VPN)

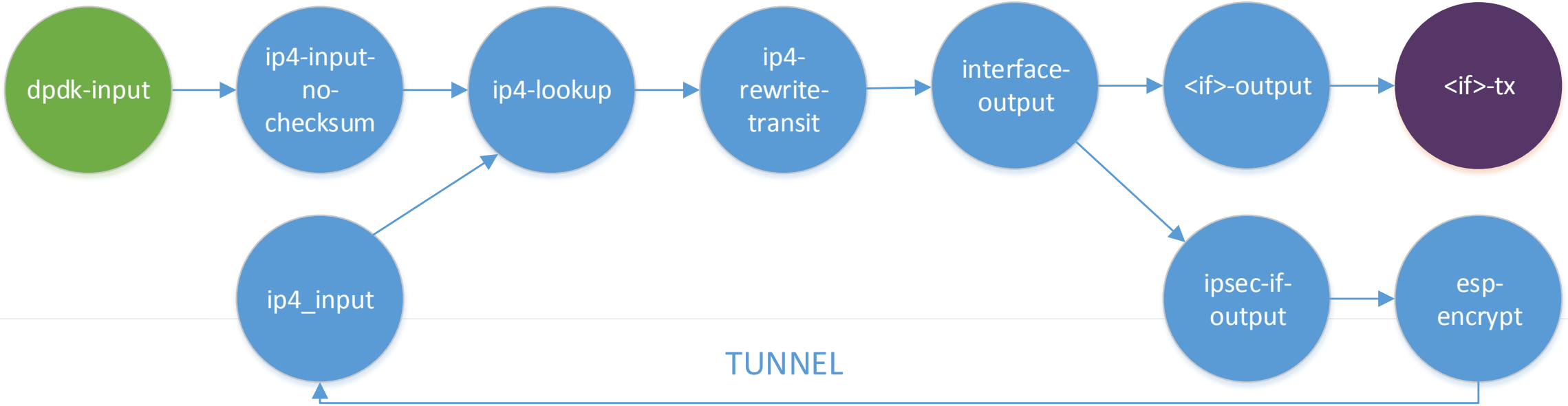
VPP 16.09

- L2GRE over IPSec

VPP 17.01

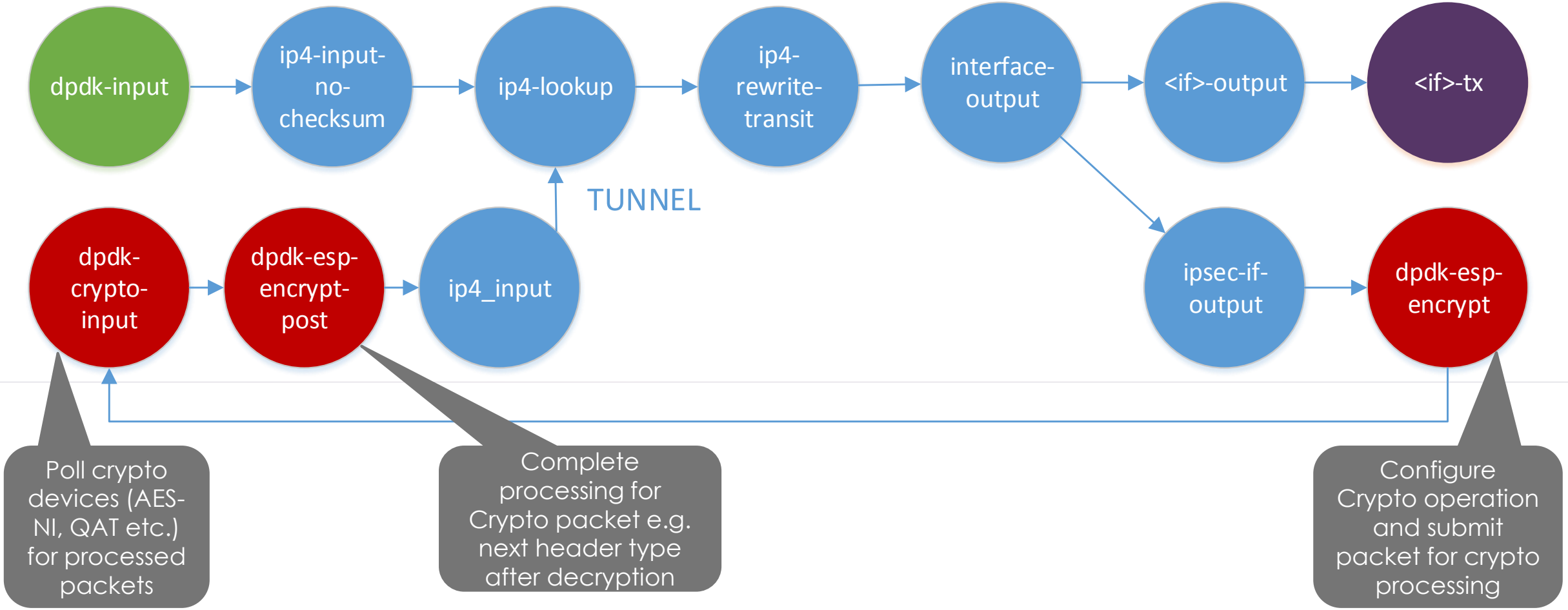
- Enable Cryptodev
- AES-GCM
- Dynamic Anti-Replay Window

Enabling Cryptodev in FD.IO/VPP (1)



VPP IPSec interface graph – outbound path

Enabling Cryptodev in FD.IO/VPP (2)



Test Setup

- Intel® Xeon® DP-based Server (2 CPU sockets).
- Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz (Haswell)
- 18 physical cores per CPU (i.e. per socket)
- 128 GB DDR4 RDIMM Crucial Server capacity = 64 GB RAM (16 x 8 GB). Tested with 128 G
- 1 x Intel® 82599 10 Gigabit Ethernet Controller
- 1 x Intel Corporation DH895XCC Series Intel® QuickAssist Technology (Coletto Creek)
- Operating System: Ubuntu 16.04, Kernel version: 4.4.0-22-generic
- VPP commit ID: 154d445f7f8f1553d9bb00d1be42bf1b06eda9f1
- Intel(R) DPDK 16.04
- Single data processing core
- All hardware local to socket

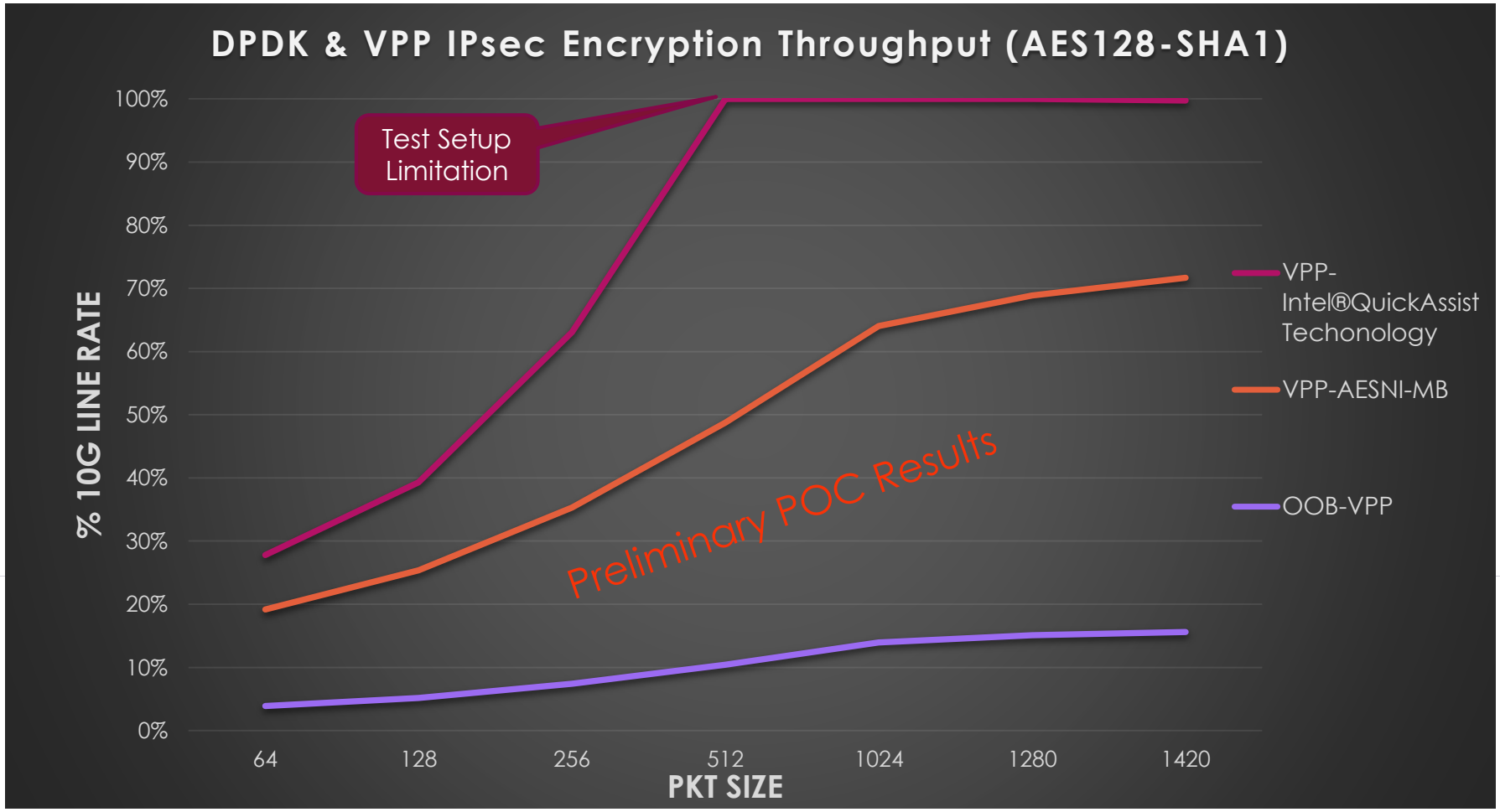
BIOS Settings	Setting
Enhanced Intel SpeedStep®	DISABLED
Processor C3	DISABLED
Processor C6	DISABLED
Intel® Hyper-Threading Technology (HTT)	DISABLED
Intel® Virtualization Technology	ENABLED
Intel® Virtualization Technology for Directed I/O (VT-d)	DISABLED
MLC Streamer	ENABLED
MLC Spatial Prefetcher	ENABLED
DCU Data Prefetcher	ENABLED
DCU Instruction Prefetcher	ENABLED
Direct Cache Access (DCA)	ENABLED
CPU Power and Performance Policy	Performance
Memory Power Optimization	Performance Optimized
Intel® Turbo boost	OFF
Memory RAS and Performance Configuration -> NUMA Optimized	ENABLED

Results will vary depending on software, workloads and system configuration

VPP Configuration


```
set int ip address TenGigabitEthernet86/0/0 192.168.10.1/24
set int promiscuous on TenGigabitEthernet86/0/1

set int ip address TenGigabitEthernet86/0/1 192.168.1.1/24
set int promiscuous on TenGigabitEthernet86/0/1

create ipsec tunnel local-ip 192.168.1.1 local-spi 1111 remote-ip 192.168.1.2 remote-spi 2222

set interface ipsec key ipsec0 local crypto aes-cbc-128 2b7e151628aed2a6abf7158809cf4f3d
set interface ipsec key ipsec0 local integ sha1-96 68676665686766656867666568676665686766656867666568676669
set interface ipsec key ipsec0 remote crypto aes-cbc-128 2b7e151628aed2a6abf7158809cf4f3d
set interface ipsec key ipsec0 remote integ sha1-96 68676665686766656867666568676665686766656867666568676669

ip route add 192.168.20.2/32 via ipsec0

set ip arp TenGigabitEthernet86/0/0 192.168.1.2 90:e2:ba:b0:dc:69

set int state TenGigabitEthernet86/0/1 up
set int state TenGigabitEthernet86/0/0 up
set int state ipsec0 up
```

Early Development Performance



Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.
All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Future Work

- Cryptodev Load-Balancer/Scheduler
- Crypto PMDs improvements
- Support for more crypto algorithms
- Scatter-Gather List Support

Questions?

Sergio González Monroy

sergio.gonzalez.monroy@intel.com

Legal Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. **No computer system can be absolutely secure**. Check with your system manufacturer or retailer or learn more at intel.com.