Augmenting P4-DPDK software pipelines with accelerators: the IPsec use-case

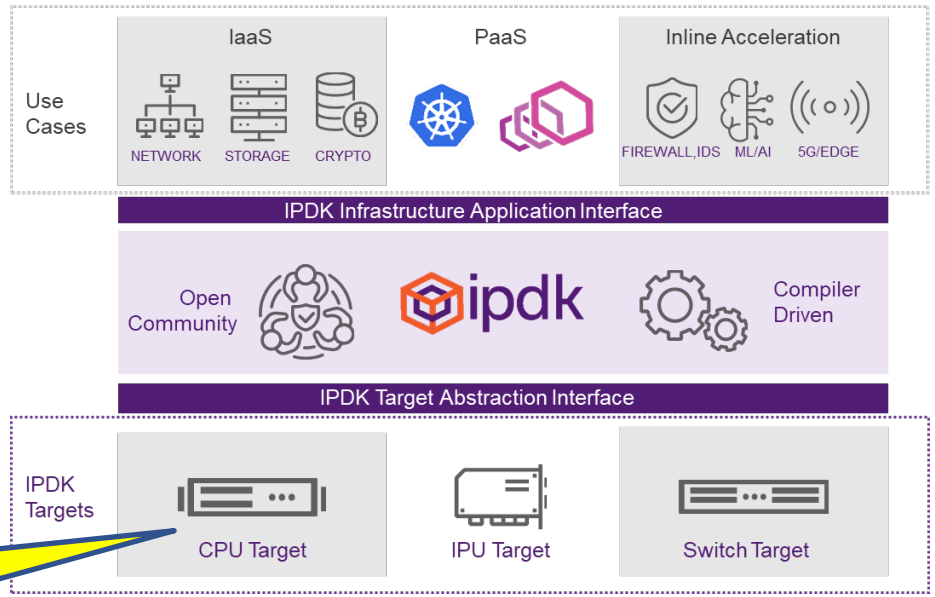
Cristian Dumitrescu, Usha Gupta, Venkata Suresh Kumar P,
Kamalakannan R, Yogesh Jangra, Harshad Narayane, Andy Fingerhut
Intel

Agenda

1. P4-DPDK Big Picture
2. P4-DPDK Feature Update
3. P4-DPDK IPsec Overview
4. Conclusions

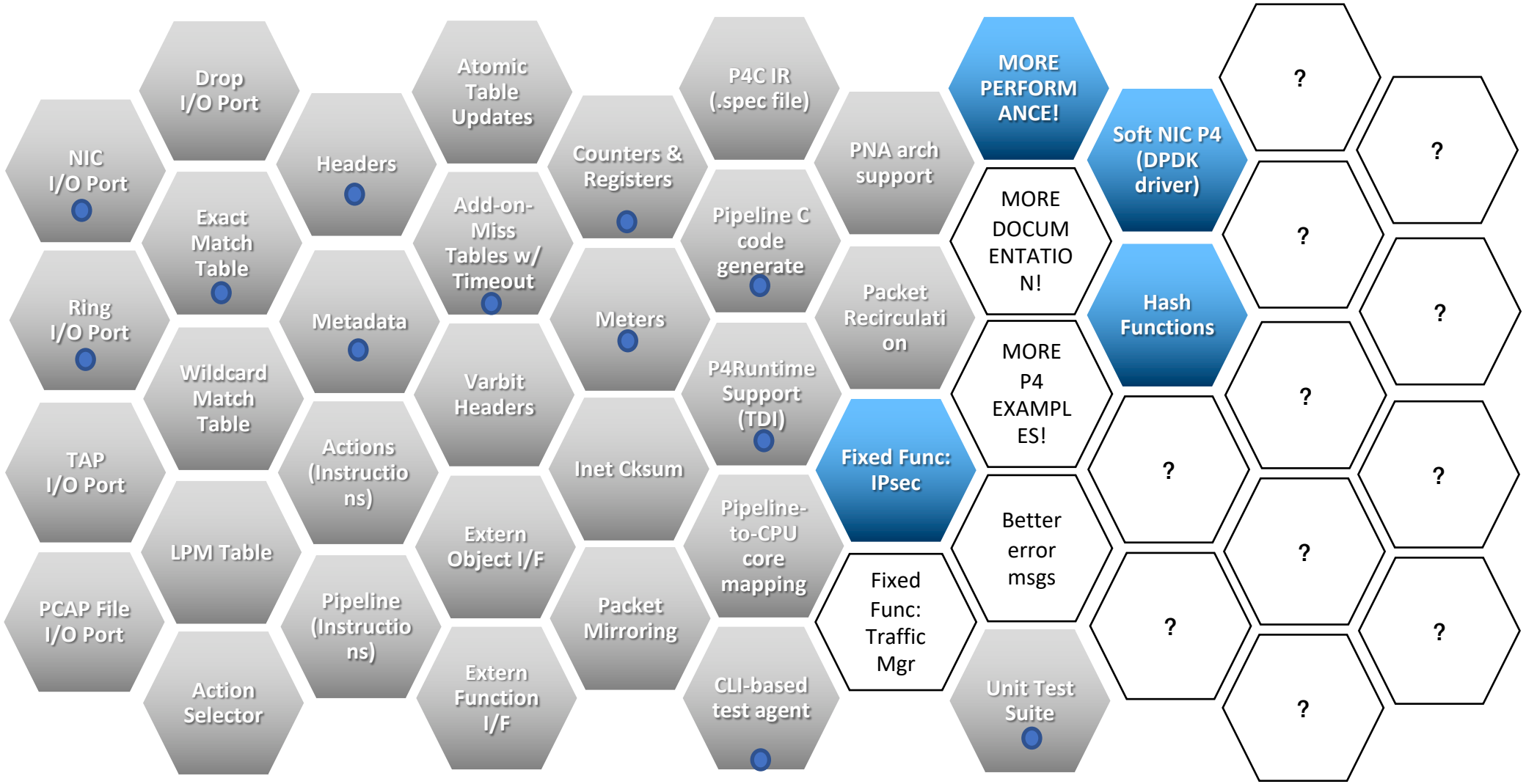
P4-DPDK: What is it

- Open-source framework to run P4 programs on multi-core CPUs.
- Goal: Develop better and faster SW switches and network stacks by combining the P4 language flexibility with the DPDK performance.
- The IPDK project uses P4-DPDK as the CPU target.
- Open-source:
  - P4 compiler back-end and TDI driver on p4.org
  - P4 data plane engine on dpdk.org.



P4-DPDK is getting better, faster and more pervasive every year!

P4-DPDK Feature Update (since P4 Workshop 2022)

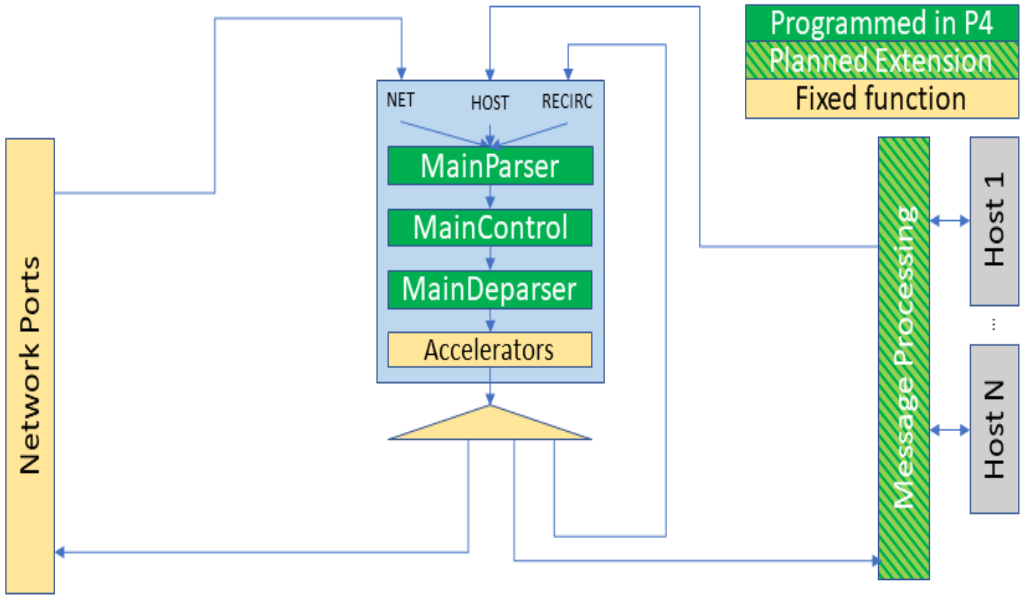


P4-DPDK Feature Update (since P4 Workshop 2022) (2)

Item (new/existing)	Details
IPsec	Fixed function pipeline companion for IPsec.
Soft NIC (DPDK driver)	The Soft NIC device driver can now run a P4 program (translated to pipeline.so first): https://git.dpdk.org/dpdk/tree/drivers/net/softnic
Hash functions	Specified in PSA, non-cryptographic digest over n-tuple for load balancing, flow affinity, etc.
I/O ports	Added non-blocking behavior.
Exact Match Table	Added support for configurable hash function for table bucket compute (e.g. jhash, CRC hash, etc).
Headers/Meta-data	Added support for large fields (field size > 64 bits).
Add-on-Miss Tables	Reworked the timer mechanism: Explicit table key re-arm on table lookup hit instead of automatic.
Counters, Registers, Meters	Added support for direct counters, registers, meters; previously, only indirect counters, registers, meters were supported. Limitation: direct counters, registers, meters supported for exact match tables and add-on-miss tables, but not supported for wildcard match tables and LPM tables.
Pipeline C code generate	The build process (pipeline.p4 -> pipeline.spec -> pipeline.c -> pipeline.so) can now be customized by the user. The legacy “interpreted” mode (executing the pipeline.spec file directly without prior translation to .c and .so) removed. DONE AS PROMISSED LAST YEAR! ☺

Augmenting P4 pipelines with accelerators

- Pipelines:
  - Programmable => Can do many things
- Accelerators/extern blocks:
  - Fixed function (configurable, but not programmable) => It only does one thing
  - HW pipelines: accelerators are HW blocks stitched at design time
  - SW pipelines: accelerators are SW blocks (reusable, scalable) stitched in flexible ways at app init time
    - Parallel execution: Executed in parallel with the pipelines on the same or different CPU core (multi-core CPUs)
    - Async comm with the pipeline: Pipeline does not need to wait for the accelerator to complete



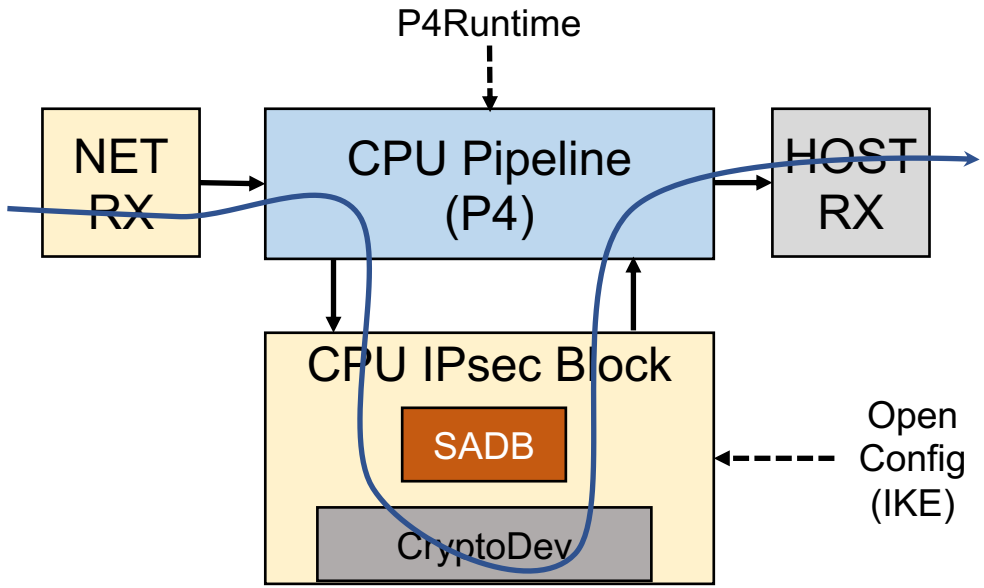
Portable NIC Architecture (PNA)

P4-DPDK: IPsec key points

- Supports the IPsec inbound and outbound processing in tunnel and transport modes.
- Reuses the DPDK IPsec and crypto libraries for the underlying implementation of the cryptographic ciphers and hashes.
- A set of purposely crafted externs hide the complexity of using the IPsec block away from the P4 program developer.

P4-DPDK: IPsec inbound path

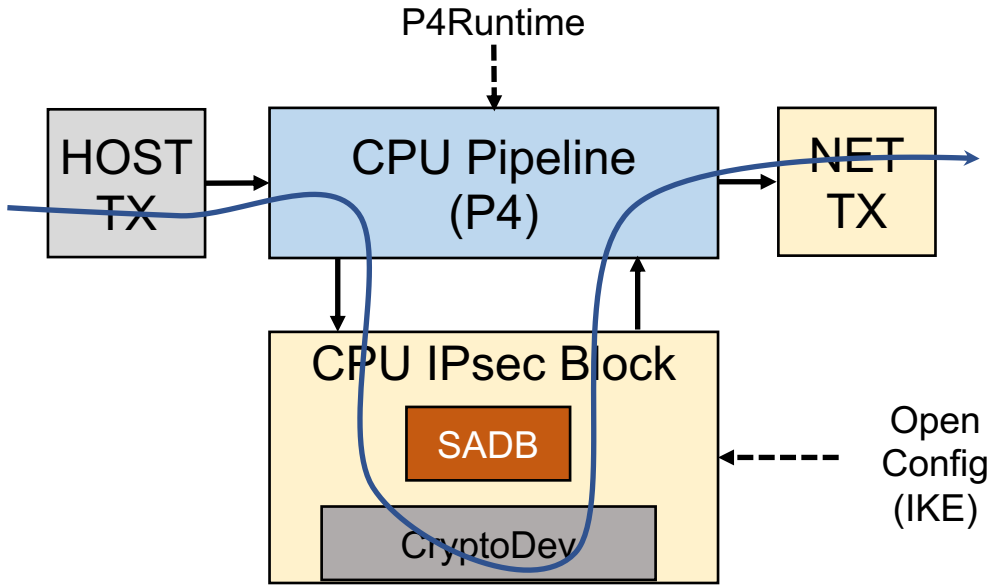
1. The pipeline reads pkt from NET RX (encrypted Ethernet pkt):
   - The Ethernet header is removed.
   - An intrinsic header containing the SA id is prepended to the IP header.
   - The pkt is sent out on a reserved output port connected to the IPsec block.
   - Externs: ipsec.from_ipsec(status) == FALSE, ipsec.enable(), ipsec.set_sa_index(sa_id).
2. The IPsec block reads IP pkt (encrypted) with intrinsic header:
   - The SA id is read from the intrinsic header, which gets consumed.
   - The pkt is decrypted based on SA indicated by the SA id. The ESP header and trailer and the outer IP header (for tunnel mode) are removed.
   - The pkt is sent pack to the pipeline on reserved input port.
3. The pipeline reads pkt from the IPsec block (decrypted IP pkt):
   - Pkts with decrypt error are dropped.
   - Pkts decrypted successfully get Ethernet header and sent to a HOST RX port.
   - Externs: ipsec.from_ipsec(status) == TRUE.



Link to the IPsec externs (ipsec_accelerator.p4) and the sample IPsec program (ipsec.p4):
https://github.com/ipdk-io/networking-recipe/tree/main/p4src/Inline_IPsec

P4-DPDK: IPsec outbound path

1. The pipeline reads pkt from HOST RX (clear text Ethernet pkt):
   - The Ethernet header is removed.
   - An intrinsic header containing the SA id is prepended to the IP header.
   - The pkt is sent out on a reserved output port connected to the IPsec block.
   - Externs: ipsec.from_ipsec(status) == FALSE, ipsec.enable(), ipsec.set_sa_index(sa_id).
2. The IPsec block reads IP pkt (clear text) with intrinsic header:
   - The SA id is read from the intrinsic header, which gets consumed.
   - The pkt is encrypted based on SA indicated by the SA id. The ESP header and trailer and the outer IP header (for tunnel mode) are added.
   - The pkt is sent pack to the pipeline on reserved input port.
3. The pipeline reads pkt from the IPsec block (encrypted IP pkt):
   - Pkts with encrypt error are dropped.
   - Pkts encrypted successfully get Ethernet header and sent to a NET TX port.
   - Externs: ipsec.from_ipsec(status) == TRUE.



Link to the IPsec externs (ipsec_accelerator.p4) and the sample IPsec program (ipsec.p4):
https://github.com/ipdk-io/networking-recipe/tree/main/p4src/Inline_IPsec

Conclusions

1. P4-DPDK has great functional coverage versus the P4, PSA and PNA specs! There are limitations, but they represent the exception, not the rule.
2. P4-DPDK can be used for the rapid development of complex CPU network stacks that also require the IPsec processing.
3. P4-DPDK is becoming better, faster and more pervasive every year!

Thank You!