Technology Guide

Intel® AVX-512 - High Performance IPsec with Intel® Xeon® Scalable Processor

Authors

Roy Fan Zhang

Georgii Tkachuk

Maciek Konstantynowicz

Pablo De Lara Guarch

Tomasz Kantecki

1 Introduction

In April 2021, we published the step-by-step guide to achieve 1 Terabit per second (Tbps) internet protocol security (IPsec) throughput with Fd.io Vector Packet Processing (VPP) running on a 3rd Gen Intel Xeon® Scalable processor. 4th Gen Intel® Xeon® Scalable processors have many impressive hardware specification improvements compared to 3rd Gen Intel® Xeon® Scalable processors. However, how do the on-paper improvements translate to the performance of a real network processing workload? Or, in other words, does Moore's Law still hold in terms of performance improvement towards the same application?

To answer this question, we ran the same tests to measure the maximum Non-Drop-Rate (NDR) IPsec throughput on the dual-socket 4th Gen Intel® Xeon® Scalable processor to find out how much performance improvement is achieved with the latest hardware and software advancements. Spoiler alert, the performance improvement is over 80%![1]

This guide describes step-by-step how we achieved the amazing up to 1.89 Tbps bidirectional IPsec tunnel NDR performance in the latest 4th Gen Intel Xeon Scalable processor. We also describe the software and hardware optimization done to achieve this performance.

This document is intended for communication service providers, or anyone looking to improve their IPsec throughput in their network. Even though the goal of this document is to showcase up to 1.89 Tbps, the technologies enabled here can be used as reference points for improving performance in any IPsec or networking deployment.

This document is part of the Network & Edge Platform Experience Kits.

[1] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	January 2023	Initial release.
002	April 2023	Clarified number of worker cores per socket in test setup.

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AES-GCM	Advanced Encryption Standard Galois/Counter Mode
CLI	Command Line Interface
CRB	Customer Reference Board
DDP	Dynamic Device Personalization (DDP)
DSCP	Differentiated Services Code Point
ESP	Encapsulating Security Payload
FD.io	Fast Data Input/Output
GRE	Generic Routing Encapsulation
IKE	Internet Key Exchange
Intel® AES-NI	Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI)
Intel® AVX-512	Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
IPC	Instruction per Cycle
IPsec	Internet Protocol Security
IPsec-MB	Intel® Multi-Buffer Crypto for IPSec
MAC	Media Access Control
NDR	Non-Drop Rate
NFV	Network Functions Virtualization
PCIe	Peripheral Component Interconnect Express
RSS	Receiver Side Scaling
RX	Receive
SA	Security Association
SP	Security Policy
SPD	Security Policy Database
SR-IOV	Single Root Input/Output Virtualization
TDP	Thermal Design Power
TX	Transmit
UDP	User Datagram Protocol
UPI	Ultra Path Interconnect
VAES	Vectorized Advanced Encryption Standard
vAPI	Virtual Application Programming Interface
VPP	Vector Packet Processing

1.2 Reference Documentation

Table 2. Reference Documents

Reference	Source
Intel AVX-512 Overview	https://www.intel.com/content/www/us/en/architecture-and-technology/avx-512-overview.html
Intel® Ethernet Network Adapter E810-2CQDA2 Overview	https://cdrdv2.intel.com/v1/dl/getContent/639389
VPP Wiki	https://wiki.fd.io/view/VPP
VPP Crypto Infrastructure and VPP IPSec Overview	https://wiki.fd.io/view/VPP/IPSec_and_IKEv2

Reference	Source
3rd Generation Intel® Xeon® Scalable Processor - Achieving 1 Tbps IPsec with Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Technology Guide	https://networkbuilders.intel.com/solutionslibrary/3rd-generation-intel-xeon-scalable-processor-achieving-1-tbps-ipsec-with-intel-advanced-vector-extensions-512-technology-guide
Intel® Speed Select Technology (Intel® SST)	https://www.intel.com/content/www/us/en/architecture-and-technology/speed-select-technology-article.html
Fast Multi-buffer IPsec Implementations on Intel® Architecture Processors	https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/fast-multi-buffer-ipsec-implementations-ia-processors-paper.pdf
Crypto Acceleration: Enabling a Path to the Future of Computing	https://newsroom.intel.com/articles/crypto-acceleration-enabling-path-future-computing/#gs.n6t02b
Intel® Multi-Buffer Crypto for IPSec	https://github.com/intel/intel-ipsec-mb
VPP/Command-line Interface (CLI) Guide	https://wiki.fd.io/view/VPP/Command-line_Interface_(CLI)_Guide
A Universal Terabit Network Dataplane	https://www.youtube.com/watch?v=aLJ0XLeV3V4

2 Overview

Compared to the 3rd Gen Intel Xeon® Scalable processor, the 4th Gen Intel® Xeon® Scalable processor has quite a few hardware specification improvements relevant to network application processing. All specifications on paper look intriguing, but the percentage of improvement, such as the instruction per cycle (IPC) gen-to-gen improvement, does not necessarily translate to the percentage of the application performance improvement. We already reached record-high system-wide 1 Tbps IPsec throughput on 3rd Gen Intel Xeon Scalable processor. With this next gen CPU, how much more can we push the record with more core count, better IPC, bigger L3 cache, and much bigger PCIe bandwidth? In other words, does Moore's Law still hold in terms of software performance? We cannot wait to find out.

2.1 Technology Description

2.1.1 4th Gen Intel Xeon Scalable Processor

The new 4th Gen Intel Xeon Scalable processor, based on Intel 7 process node, is a revolutionary computer platform designed for workload acceleration. It has a maximum of 60 CPU cores, which is 50% more than 3rd Gen Intel Xeon Scalable processor, not to mention each CPU core has up to 15% IPC improvement over 3rd Gen[2]. It offers 8-channel 4800MT/s DDR5 memory, which is more than 1.5 times memory bandwidth than 3rd Gen Intel Xeon Scalable processor. Moreover, its 80 PCIe 5.0 lanes provide more than 100% extra PCIe bandwidth over last gen processors.

The 4th Gen Intel Xeon Scalable processor also inherits the Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instruction accelerations. The relevant encryption instructions to boost cryptographic operation performance, such as VPMADD52 (vector instruction that does integer multiply accumulate), vAES (vector version of the Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) instructions), vPCLMUL (vectorized carry-less multiply), and Intel® Secure Hash Algorithm - New Instructions (Intel® SHA-NI) (secure hash algorithm new instructions) now are further enhanced by the 15% IPC performance improvement of general purpose instructions over the 3rd Gen Intel® Xeon® Scalable processor[2].

Refer to the following for an overview of key new technologies:

- New Intel AVX-512 instruction set support for accelerated processing of vectorized instructions.
  For more information on Intel AVX-512, refer to: Accelerate Your Compute-Intensive Workloads: Intel® Advanced Vector Extensions 512 (Intel® AVX-512)

- New Intel® Speed Select Technology (Intel® SST) power management technologies for increased power-aware performance.
  For more information on Intel SST, refer to: Meet Intel® Speed Select Technology (Intel® SST)

2.1.2 Intel® Ethernet 800 Series Network Adapter

Intel® Ethernet Network Adapter E810-2CQDA2 delivers up to 200 Gbps of total bandwidth in systems[3] that are PCIe 4.0 compliant. Each QSFP28 port supports up to 100 Gbps, providing the functionality and throughput of two 100 Gbps adapters in a single bifurcated PCIe 4.0 x16 slot. It is designed for optimizing networking workloads including network functions virtualization (NFV) and features technologies such as the following:

[2] Based on SPECInt2017 benchmark measurements on 3rd and 4th Gen Xeon Scalable processors at same core count, same frequency, same compiler

[3] Requires x16 slot bifurcation

- Intelligent Flow Direction: Receiver Side Scaling (RSS).
- Comprehensive Network Virtualization Overlay Protocols Support.
- vSwitch Assist.
- QoS: Priority-based Flow Control (802.1Qbb).
- Enhanced Transmission Selection (802.1Qaz).
- Differentiated Services Code Point (DSCP).
- Dynamic Device Personalization (DDP).

For more information about Intel® Ethernet Network Adapter E810-2CQDA2, refer to: Intel® Ethernet Network Adapter E810-2CQDA2.

2.1.3 Intel® Multi-Buffer Crypto for IPsec Library

The Intel® Multi-Buffer Crypto for IPSec library is a family of highly optimized software implementations of the symmetric cryptographic algorithms. With the rich and easy-to-use APIs provided by the Intel Multi-Buffer Crypto for IPSec library, you can easily make full use of the latest cryptographic accelerations provided by Intel CPUs, including the new vAES and vPCLMUL instructions. These Intel AVX-512-accelerated instructions allow processing up to four 128-bit AES blocks in parallel, getting theoretically up to four times better performance than the 3rd Gen Intel Xeon Scalable processor. Moreover, the Intel Multi-Buffer Crypto for IPSec library hides all implementation details to accommodate different CPU flags (SSE, AVX, AVX2, AVX512) behind the APIs, which ensures highly optimized cryptographic operation results for all Intel® CPUs in the market and provides the user seamless transition of their code into 3rd Gen Intel Xeon Scalable processor CPU based systems.

For more detailed information about the Intel Multi-Buffer Crypto for IPSec library and Intel vAES and vPCLMUL instructions, refer to:

- Fast Multi-buffer IPsec Implementations on Intel® Architecture Processors White Paper
- Crypto Acceleration: Enabling a Path to the Future of Computing
- Intel® Multi-Buffer Crypto for IPSec

2.1.4 Fast Data Input/Output (FD.io), Vector Packet Processing (VPP)

FD.io (Fast Data Input/Output) is a Linux Foundation open-source project that provides high throughput network packet processing capabilities. FD.io Vector Packet Processing (VPP) is one of the many sub-projects within FD.io that provides L2-L4 stack processing.

The term “vector” in VPP is essentially a group of packets, referred to as a “vector of packets” or a “packet vector”. Each function block treats a packet vector, up to a maximum of 256 packets, as an input and processes them in an identical manner. This helps maximize highly efficient utilization of CPU instruction cache (I-cache). In addition, VPP innovatively adopts a Packet Processing Graph as part of its core design, in which case each function block is abstracted as a graph node. The graph nodes are organized as a tree shape graph by registering the “next” output nodes either initially or at runtime. The packet vectors flow from the Ethernet Adapter Receive (RX) nodes all the way to Transmit (TX) nodes (or dropped) based on the processed destinations in each graph node within.

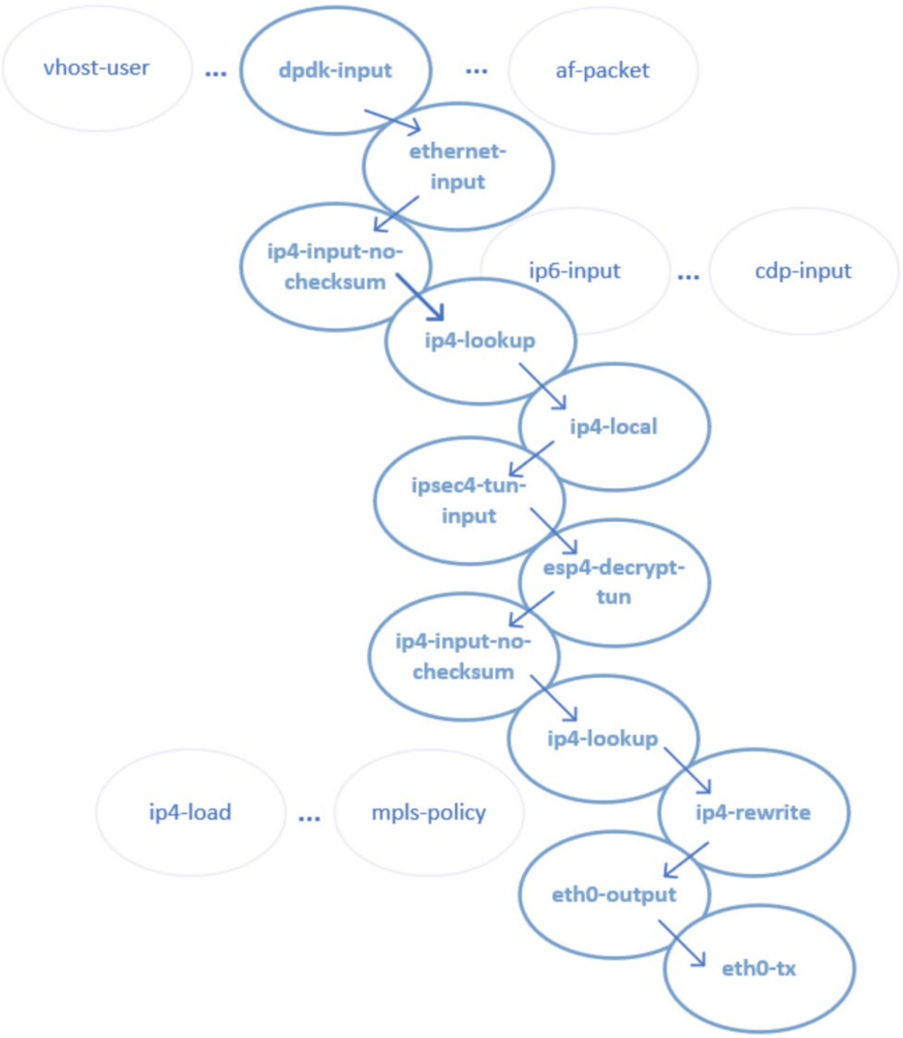


Figure 1. VPP Packet Processing Graph with IPSec-related nodes highlighted

The packet processing graph has the distinct advantage of extreme flexibility. New graph nodes can be “plugged in” anywhere and existing ones can be bypassed via simple software or real-time command line configuration. It is also efficient. The run-to-completion design allows the vector of packets to remain in the data cache for the entire duration of the packet processing pipeline.

DPDK plugin for VPP allows VPP to take advantage of all optimizations and hardware enablement features like NIC and hardware accelerator drivers included in DPDK. This includes the poll mode driver for the E810-series NICs optimized with AVX512 intrinsics for vectorizing packet receive and transmit operations.

For more information about VPP, refer to:

- FD.io Wiki
- VPP Configuration File - ‘startup.conf’ — Vector Packet Processor 0.1 documentation (my-vpp-docs.readthedocs.io)
- Getting Started with the debug CLI — The Vector Packet Processor v23.02-rc0-198-g57f177d0b documentation (fd.io)
- VPP/IPSec - fd.io

2.1.5 VPP IPsec

VPP IPsec is an important component that serves as a basis in VPP to provide secure, reliable, and fast networking applications. VPP IPsec provides a set of easy-to-use command line interface (CLI) and virtual application programming interface (vAPI) commands for user to configure the security policy database (SPD), security associations (SA), and associated cryptographic algorithms and keys.

VPP IPsec supports:

- Major cipher, authentication, and AEAD cryptographic algorithms including:
  - Cipher: AES-CBC-128/192,256, AES-CTR-128/192/256

  - Authentication: HMAC-MD5, HMAC-SHA-96/224/256/384/512
  - AEAD: AES-GCM-128/192/256, ChaCha20-Poly1305
- ESP tunnel and transport mode, optional over UDP or GRE
- Authentication header
- IKEv2 initiator and responder

The most resource intensive procedure within IPsec is the cryptographic operation. To ensure both the performance and the flexibility of the cryptographic operations, VPP IPsec takes advantage of underlying crypto infrastructure.

2.1.6 VPP Crypto Infrastructure and Engines

The VPP crypto infrastructure is a crypto framework that supports different crypto engines working as plugins for high performance symmetric crypto operations. At the time of this writing, there are three crypto engines:

- Native engine: The crypto engine that is specifically designed for VPP that achieves the fastest crypto processing efficiency but with limited algorithms supported. vAES and vPCLMUL acceleration of AES encryption/decryption are automatically enabled if the application is running on the latest x86 architecture CPUs.

- IPSec-MB engine: Integration layer to Intel® Multi-Buffer Crypto for IPSec library with extended crypto algorithm support list but slightly less performance compared to the native engine. vAES and vPCLMUL acceleration of AES encryption/decryption are automatically enabled if the application is running on the latest x86 architecture CPUs.

- OpenSSL engine: The shim-layer to OpenSSL library with the most comprehensive crypto algorithm support list but is least performant.

The VPP crypto infrastructure provides a high-level API for all VPP components. Underneath the APIs the default crypto engine that handles the specific algorithms' operation is invoked to process the crypto operation. This flexible operation mode ensures that the most performant crypto implementation is used for a specific algorithm.

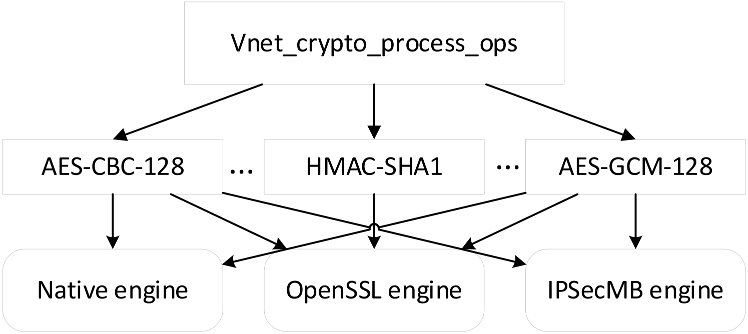


Figure 2. VPP Crypto Infrastructure

3 Deployment

In this section we explain the hardware and software configuration that was used to achieve the target throughput. There are many ways to set up VPP IPsec on a server, and the configuration presented in Table 3 is one of many possible ways to achieve optimal IPsec performance. Some variations of this configuration may be required to enable certain functionality and may result in better or worse performance. The system BIOS settings for our test can be found in Appendix A.

Table 3. System Setup and Software Versions

Item	Description
Server Platform	Intel Archer City Customer Reference Board (CRB)
Test by	Intel as of Nov 2022
CPU	2 Sockets Intel® Xeon® Platinum 8470N CPU @ 1.70 GHz
Memory	Manufacturer: Hynix, Speed: 4800 MT/s, Number: 8 per socket, 1 DIMM Per Channel
Ethernet Adapter	10x Intel® Ethernet Network Adapter E810-2CQDA2

Item	Description
Ethernet Adapter Driver Version	Ice 1.8.9
Ethernet Adapter Firmware Version	4.00
BIOS	EGSDCRB1.86B.8901.P01.2209200239
Microcode	0xab0000c0
Operating System	Ubuntu 22.04 (Jammy Jellyfish)
Linux Kernel Version	5.15.35 (built from source with CONFIG_NO_HZ_FULL=y)
Added kernel GRUB command options	hugepagesz=2M hugepages=8192 default_hugepagesz=2M isolcpus=1-51,53-103,105-155,157-207 rcu_nocbs=1-51,53-103,105-155,157-207 nohz_full=1-51,53-103,105-155,157-207 intel_pstate=enable rcu_nocb_poll skew_tick=1 nomodeset clocksource=tsc kthread_cpus=0 irqaffinity=0 powernow-k8.tscsync=1
VPP Version	22.02-release

3.1 Hardware Components

The 4th Gen Intel Xeon Scalable processor CPU comes in many flavors of different core counts, base CPU frequencies, TDP, and other factors. For our test we chose one of the networking SKU offerings: Intel® Xeon® Platinum 8470N processor. This CPU has 52 cores, 1.7 GHz base, and all-core Turbo CPU frequency of 2.7 GHz. A single 4th Gen Intel Xeon Scalable processor CPU supports up to 80 Gen5 PCIe lanes, which allows us to populate five Ethernet Adapters on a single socket and 10 Ethernet Adapters when using a dual socket system. In this test we used Intel® Ethernet Network Adapter E810-2CQDA2 PCIe Gen4 x16 that can achieve up to 200 Gb/s bidirectional traffic with two 100 GbE ports. These Ethernet Adapters require that the system support PCIe bifurcation into x8x8 configuration. It is worth mentioning that at the time of this testing no PCIe Gen5 NICs were available, so we used PCIe Gen4 NICs instead, essentially using only half of IO capability of the 4th Gen Intel Xeon Scalable platform.

3.2 Test Topology

3.2.1 CPUs, Ethernet Adapters, and NUMA

To demonstrate performance of the 4th Gen Intel Xeon Scalable platform, we configured the IPSec test such that the same system processes each packet twice: encrypting the clear text packet as it arrives to the system, then decrypting it after it is moved to the other socket through the secure tunnel. Essentially, the same platform acts as both terminals of the same IPSec tunnel – each terminal running exclusively on its own CPU socket as shown in Figure 3.

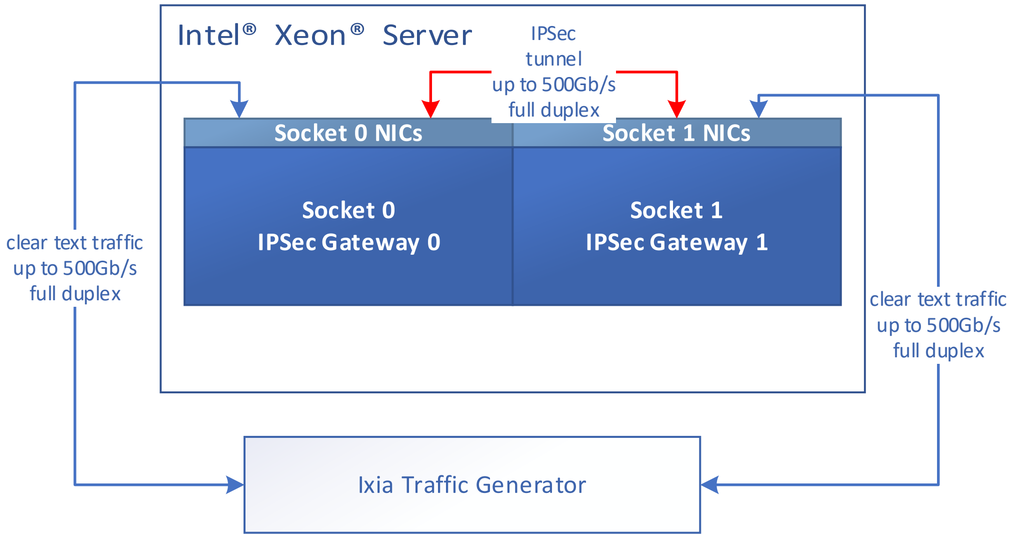

Intel® Xeon® Server
IPSec tunnel up to 500Gb/s full duplex
Socket 0 NICs
Socket 1 NICs
Socket 0 IPSec Gateway 0
Socket 1 IPSec Gateway 1
clear text traffic up to 500Gb/s full duplex
clear text traffic up to 500Gb/s full duplex
Ixia Traffic Generator

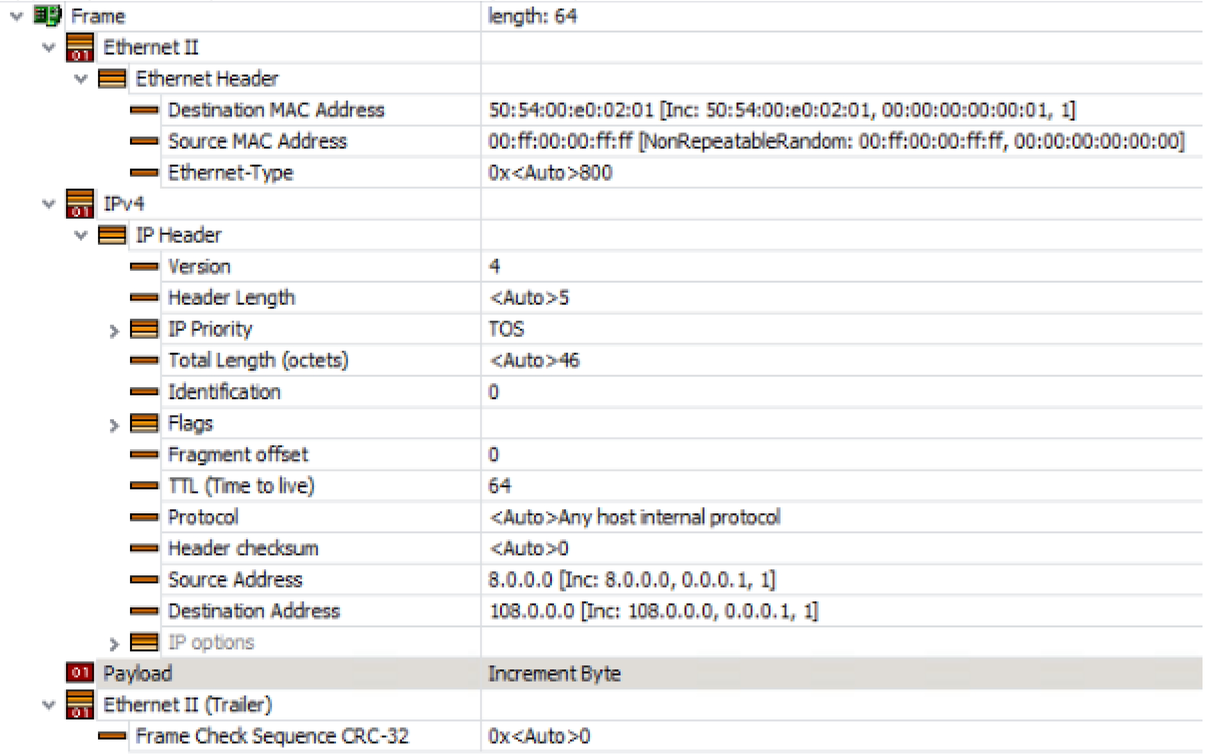
Figure 3. Simplified Connection Diagram

We deployed two instances of VPP application on each CPU socket and configured them as two IPsec gateways with statically configured IPsec tunnels between them. A minimum of two instances of VPP application per socket were required because one instance supports up to 32 Ethernet ports, which was not sufficient for our planned test configuration. Each VPP instance was configured only to use CPU cores, memory, and Ethernet Adapters from its native NUMA node to avoid long latency memory operations moving data from one NUMA node to another across Ultra Path Interconnect (UPI).

Half of all 100 GbE Ethernet Adapter ports were connected to a Keysight Ixia traffic generator and hardware traffic generator and the remaining ports were connected to their IPsec tunnel counterparts. The Ixia hardware traffic generator was configured to send clear text network packets to each gateway. We created one IPsec tunnel between each pair of interconnected Ethernet Adapters and configured the Ethernet Adapters to use SR-IOV virtual functions, each of which was driven by a dedicated CPU core. As a result, each IPsec tunnel was driven by a dedicated pair of cores, one of which encrypted outbound IPsec traffic while the other decrypted the inbound traffic. With four virtual functions per Ethernet port, we used a total of 40 CPU cores per socket and left remaining 12 CPU cores per socket unused. We did not use Intel® Hyper-Threading Technology (Intel® HT Technology) in this demo, nor Intel® Turbo Boost Technology as we were able to achieve our target performance without relying on these technologies.

3.2.2 Ixia Flow Configuration

We configured Ixia IxNetwork software to deliver 40 uniform networking traffic streams to the system – 20 streams to each gateway. Each stream targeted a given virtual function with a Statically configured destination MAC address. Each stream consisted of uniformly timed clear text 1420 byte IPv4 packets with destination IP addresses configured such that they match one security policy (SP) entry in one of the VPP IPsec gateways.



Frame	length: 64
Ethernet II	
Ethernet Header	
Destination MAC Address	50:54:00:e0:02:01 [Inc: 50:54:00:e0:02:01, 00:00:00:00:00:01, 1]
Source MAC Address	00:ff:00:00:ff:ff [NonRepeatableRandom: 00:ff:00:00:ff:ff, 00:00:00:00:00:00]
Ethernet-Type	0x<Auto>800
IPv4	
IP Header	
Version	4
Header Length	<Auto>5
IP Priority	TOS
Total Length (octets)	<Auto>46
Identification	0
Flags	
Fragment offset	0
TTL (Time to live)	64
Protocol	<Auto>Any host internal protocol
Header checksum	<Auto>0
Source Address	8.0.0.0 [Inc: 8.0.0.0, 0.0.0.1, 1]
Destination Address	108.0.0.0 [Inc: 108.0.0.0, 0.0.0.1, 1]
IP options	
Payload	Increment Byte
Ethernet II (Trailer)	
Frame Check Sequence CRC-32	0x<Auto>0

Figure 4. Packet Stream Configuration in Ixia IxNetwork

3.2.3 VPP Application Configuration

VPP uses startup configuration files and CLI configuration scripts to set up the desired networking function. The following sections describe our VPP configuration in more detail and provide snippets of the configuration files.

3.2.3.1 VPP Startup Configuration File (startup.conf)

The VPP startup configuration files allow the user to specify how hardware resources should be allocated and configured, as well as how to configure various VPP plugins. Some startup configuration options can result in a performance gain. What follows is the list of most relevant options with our comments. We used the VPP DPDK plugin as the packet processing driver.

```
cpu {
#we make sure the main core is on the same CPU socket as the worker
cores because VPP allocates memory based on numa node of the main core
     main-core 52
#we use one core per SRIOV virtual function interface and make sure
the worker cores are on the same numa node as the NIC
     corelist-workers
53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72
}

dpdk {
#disabling tx checksum offload in the NIC allows DPDK to use
vectorized TX functions in the poll mode driver
     no-tx-checksum-offload
#saves cycles by skipping segmentation logic
     no-multi-seg
     dev default{
#512 descriptors for RX and TX rings were found to be optimal values
for best performance and manageable packet loss
           num-tx-desc 512
           num-rx-desc 512
#each virtual function gets one RX and one TX queue which are driven
by a single CPU thread
           num-rx-queues 1
           num-tx-queues 1
     }
#82:01.0 device is a SRIOV virtual function of 82:00.0 physical device
we assign the first worker core to it.
     dev 0000:82:01.0
     {
           workers 0
           name eth82-1-0
     }
     dev 0000:82:01.1
     {
           workers 1
           name eth82-1-1
     }
…
}
#we use 2MB hugepages for the heap as another optimization
memory { main-heap-page-size 2M
main-heap-size 2G }
#application needed a large memory region for stats, otherwise it
crashed
statseg { size 7G }
```

Figure 5. Snippet of Most Relevant Startup File Configuration Options

3.2.3.2 VPP CLI Commands

Figure 6 shows a snippet of the VPP IPsec gateway CLI configuration for a pair of virtual function interfaces on one of the gateways. In this example, interface eth2a-1-1 is the outbound IPsec interface and interface eth5a-1-1 is its inbound IPsec counterpart.

```
comment {#enable interfaces and assign IP addresses }
set interface mtu 2024 eth2a-1-1
set interface mtu 2024 eth5a-1-1
set interface state eth2a-1-1 up
set interface state eth5a-1-1 up
set interface ip address eth2a-1-1 2.0.13.0/24
set interface ip address eth5a-1-1 192.168.13.0/24
comment {#configure static ARP }
set ip neighbor eth2a-1-1 2.0.13.1 00:11:11:11:13:11
comment {#create IPSec tunnels and SA with AES256-GCM crypto cipher }
create ipip tunnel src 242.128.0.0 dst 242.129.0.0
ipsec sa add 130 spi 242128 crypto-key
2b7e151628aed2a6abf7158809cf4f3d2b7e151628aed2a6abf7158809cf4f3d
crypto-alg aes-gcm-256
ipsec sa add 131 spi 242129 crypto-key
2b7e151628aed2a6abf7158809cf4f3d2b7e151628aed2a6abf7158809cf4f3d
crypto-alg aes-gcm-256
ipsec tunnel protect ipip7 sa-in 130 sa-out 131
comment {#assign IP to the newly created IPSec tunnel interface }
set int ip address ipip7 242.128.0.0/32
comment {#direct packets incoming from Ixia to the IPSec tunnel }
ip route add 116.0.0.0/32 via ipip7
set int state ipip7 up
comment {#setup route for the inbound IPSec traffic }
ip route add count 1 016.0.0.0/32 via 2.0.13.1 eth2a-1-1
set ip neighbor eth5a-1-1 192.168.13.1 50:54:00:e1:09:01
ip route add 242.129.0.0/16 via 192.168.13.1 eth5a-1-1
comment {#use Intel IPSec multibuffer libraries as encryption engine }
set crypto handler all ipsecmb
```

Figure 6. Snippet of VPP IPsec Gateway CLI Configuration

3.3 Performance Measurement

Performance of the system with this IPsec workload is measured by executing Ixia RFC 2544 QuickTest with 0.00001% allowed packet loss. Each iteration of the RFC 2544 binary search algorithm is configured to run for 20 seconds. Ixia measures the throughput by registering every packet that it receives. Because each packet travels through the system under test twice – once through each IPsec gateway, we multiply the Ixia score by 2 to get the final throughput of the entire system. The total rate of clear text traffic being sent from traffic generator to the platform was 1 Terabit per second and each packet would be processed twice making the total theoretical maximum processing rate of the system at most 2 terabits per second.[4] This theoretical maximum is based on the full duplex throughput of both ports of all 10 Ethernet Adapters: 100 Gb/s * 2 * 10 = 2000 Gb/s. We expect the final score of the test to be slightly lower as IPsec protocol adds an overhead to the encrypted packet, but this overhead is removed before the packet reaches Ixia for accounting.

4 Results

Figure 7 shows performance in Gigabits per second of the test scenario described earlier when using different numbers of VPP worker cores.[5] As we can see from the chart, performance continues to increase as more CPU cores are added as worker cores to the application until reaching the target performance of 1894 Gb/s using 40 cores per socket. As we briefly mentioned before, this is only one of many possible configurations of this VPP IPsec workload. Reader may choose to use a different encryption algorithm, different packet size, or different encryption engine like Intel® QuickAssist Accelerator. One might also enable Intel® Turbo Boost Technology to reach the target performance with fewer cores.

[4] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

[5] Tests were performed by Intel in November 2022.

This result nearly doubles the performance of a similarly configured measurement previously described in the 3rd Generation Intel® Xeon® Scalable Processor - Achieving 1 Tbps IPsec with Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Technology Guide. Table 4 lays out the key differences between the 3rd Gen Intel Xeon Scalable processor test and this test.

Table 4. Configuration Comparison of Current and Previous Tests[6]

	3rd Gen Intel Xeon Scalable Processor	4th Gen Intel Xeon Scalable Processor
CPU	8360Y	8470N
Base CPU frequency	2.4	1.7
Thermal Design Power (TDP)	250W	300W
Encryption Algorithm	AES128-GCM	AES256-GCM
Number of worker cores per socket to reach maximum performance	24 per socket	40 per socket
Total IPsec performance of one server	1012 Gb/s	1895 Gb/s

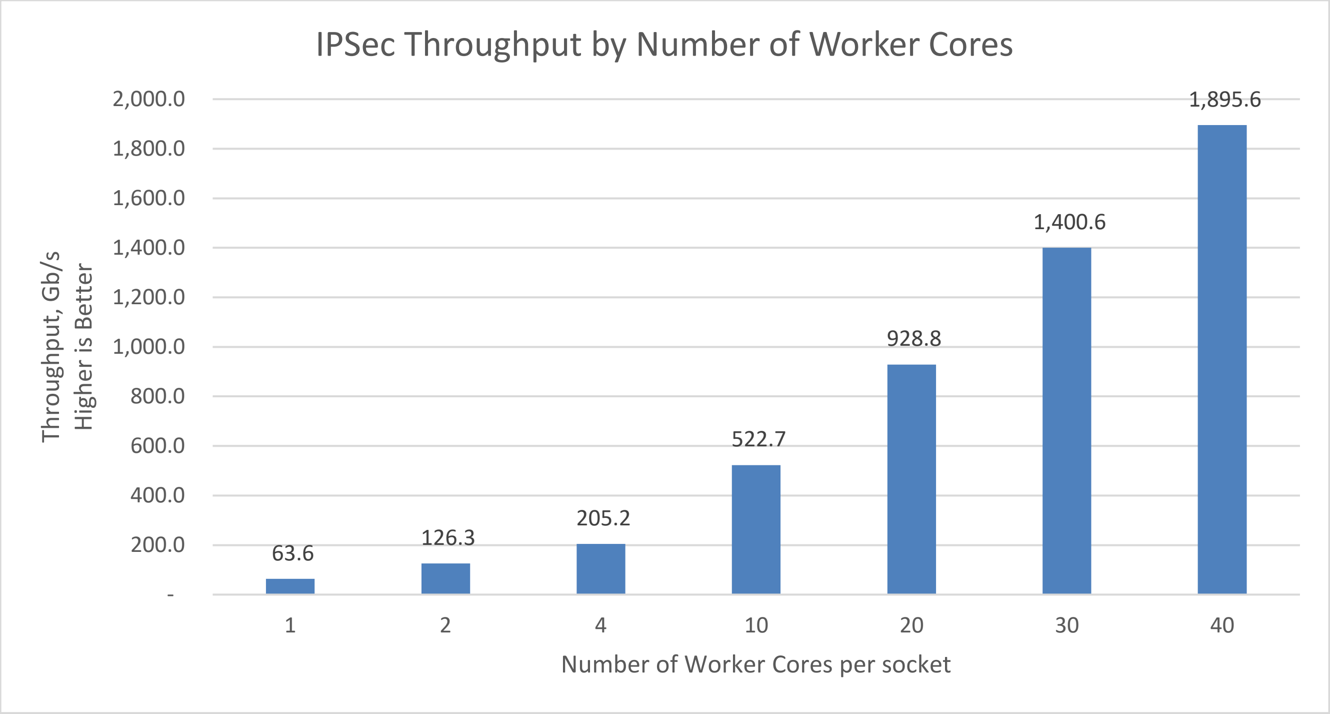


Figure 7. Performance of VPP IPsec Workload Running with Different Number of Worker Cores

5 Summary

We started to consider the idea of achieving terabit level network performance in 2016, when VPP was open sourced under the Linux Foundation FD.io. project, believing that it would likely take several years. To that end, already in 2017 we managed to reach close to 1 Tbps of throughput for clear text IPv4 routing with VPP running in a four-socket setup with the 1st Gen Intel Xeon Scalable processor (see A Universal Terabit Network Dataplane).

Advancing a few years, and as described in this paper, thanks to 4th Gen Intel Xeon Scalable processor, substantial improvements in PCIe I/O, ISA, and IPC, all critical to network packet processing, we can now get close to 2 Tbps IPsec zero frame loss network throughput on a two-socket server using 40 workers cores per socket, averaging ~25 Gbps per core.

Even though the described benchmarks have been executed on bare-metal servers and in controlled lab conditions, we strongly believe it is a major milestone for the industry and an important proof point that multi-Terabit levels of system network

[6] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

performance with encryption in software (~25 Gbps per core, without any hardware accelerators) are here and ready for production deployments.

6 Acknowledgements

The authors wish to acknowledge the FD.io VPP and CSIT contributors for a number of IPsec and crypto code optimizations that enabled achieving the IPsec performance described in this paper. Special thanks to the following people for their valuable comments and suggestions that made this work possible: Benoit Ganne, Damjan Marion, Mohammed Hawari, Vratko Polák, Maciek Konstantynowicz.

Appendix A System BIOS Settings

Table 5. System BIOS Settings

Socket Config Menu	Sub-Menu	Sub-Menu2	Setting
IIO Configuration	port configuration		x8x8 for all ports
	Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d)		Enable
Advanced Power Management Configuration	CPU P-state Control	Speed Step	Disable
		Intel® Advanced Vector Extensions (Intel® AVX) License Pre-Grant	Enable
		Intel AVX ICCP pre-grant level	512 light
	Hardware PM State Control	Hardware P-states	Native Mode
	CPU C-state Control	CPU C1 auto demotion	Disable
		CPU C1 auto undemotion	Disable
	Package C-state Control	Package C-state	C0/C1 State
	CPU Advanced PM Tuning	Uncore Freq Scaling	Enable
		Uncore Freq RAPL	Enable
	Energy perf bias	Power Performance Tuning	BIOS Controls EPB
		ENERGY_PERF_BIAS_CFG Mode	Balanced Performance
		Workload Configuration	I/O Sensitive

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

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.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

0423/DN/WIT/PDF 761755 -002US