TECHNOLOGY GUIDE

Intel Corporation

3rd Generation Intel® Xeon® Scalable Processor - Achieving 1 Tbps IPsec with Intel® Advanced Vector Extensions 512 (Intel® AVX-512)

Authors

Fan Zhang

Georgii Tkachuk

Aibhne Breathnach

Declan Doherty

Tomasz Kantecki

1 Introduction

As the industry moves to next generation 5G networks and beyond, the throughput and performance capabilities of that network and those using it will scale to the hundreds of gigabits per second. Network operators are increasingly aware of the challenges posed, and the need to transport information safely and securely within the network and to the customers.

This guide describes how the latest Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instructions and Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) enabled in the latest 3rd Generation Intel® Xeon® Scalable processor are used to achieve 1 Terabit per second (Tbps) of IPsec throughput[1].

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 1 Tbps, the technologies enabled here can be used as a reference point for improving performance in any IPsec or networking deployment.

This document is part of the Network Transformation Experience Kit, which is available at https://networkbuilders.intel.com/network-technologies/network-transformation-exp-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	February 2021	Initial release.
002	April 2021	Revised the document for public release to Intel® Network Builders.

1.1 Terminology

Table 1. Terminology

ABBREVIATION	DESCRIPTION
AES	Advanced Encryption Standard
AEAD	Authenticated Encryption with Associated Data
Intel® AES-NI	Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI)
AES-GCM	Advanced Encryption Standard Galois/Counter Mode
Intel® AVX-512	Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
FD.io	Fast Data Input/Output
IPsec	Internet Protocol Security
IPsec-MB	Intel® Multi-Buffer Crypto for IPsec Library
VAES	Vectorized Advanced Encryption Standard
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® AVX-512 – Packet Processing with Intel® AVX-512 Instruction Set Solution Brief	https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-packet-processing-with-intel-avx-512-instruction-set-solution-brief
Intel® AVX-512 – Instruction Set for Packet Processing Technology Guide	https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-instruction-set-for-packet-processing-technology-guide
Intel® AVX-512 – Writing Packet Processing Software with Intel® AVX-512 Instruction Set Technology Guide	https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-writing-packet-processing-software-with-intel-avx-512-instruction-set-technology-guide
Intel® Ethernet 800 Series Network Adapter Overview	https://ark.intel.com/content/www/us/en/ark/products/series/184846/intel-ethernet-network-adapter-e810-series.html
VPP Wiki	https://wiki.fd.io/view/VPP
VPP Crypto Infrastructure and VPP IPsec Overview	https://wiki.fd.io/view/VPP/IPSec_and_IKEv2

2 Overview

This document is intended as a guide to show how IPsec throughput may be significantly increased and how 1 Tbps of IPsec throughput may be achieved with the setup detailed in this document[2].

The document describes the technologies used for this test setup and how the test system was set up and deployed to achieve over 1 Tbps of No Drop Rate (NDR) IPsec AES-GCM-128 tunnel encryption throughput.

For the purpose of this test setup, we deployed two Device Under Test (DUTs) each with Dual Socket Xeon® Scalable processors. The hardware and software used in this test setup include:

- A system with two of the latest Intel® Xeon® Scalable processors
- Six Intel® Ethernet Network Adapter E810-2CQDA2 PCIe 4.0 NICs that deliver 200 Gbps total bandwidth populated with two Intel® Ethernet Controller E810-CAM1 chips, providing for a total of 12x 100 GbE interfaces
- Fast Data Input/Output (FD.io) Vector Packet Processing (VPP) open source high performance, packet-processing stack
- Intel® AVX-512 and Intel® AES-NI instructions for packet encryption/decryption

While this guide explains how the 1 Tbps figure was achieved, the system setup and technology enablement demonstrated in this guide are intended as a reference guide for anyone trying to improve their networking or IPsec performance at any level of throughput.

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

2.1 Challenges Addressed

To secure the network traffic with Internet standards such as IPsec, the challenge to an efficient IPsec gateway system becomes even larger:

- The high packet rate requires powerful and isolated CPU cores to perform network stack computation at high speed.
- The high data rate requires extremely efficient cryptography computation capability for both inbound and outbound protected traffic.
- The large simultaneous network flows for both encrypted and clear-text packets, requires high throughput and intelligent Network Interface Cards (NICs) to classify network packets efficiently for different CPU cores to digest and process.

The market demands fast and scalable network application solutions. To date, several software technology innovations have been adopted to address the challenge, including Software-Defined Networking (SDN) and Network Function Virtualized (NFV).

In this document, we present our solution to achieve over 1 Tbps No Drop Rate (NDR) IPsec AES-GCM-128 Tunnel Encryption/Decryption throughput processing capability with Intel® Ethernet Network Adapter E810-2CQDA2 and 3rd Generation Intel® Xeon Scalable processor[3].

2.2 Technology Description

2.2.1 3rd Generation Intel® Xeon® Scalable Processor

The latest Intel® Xeon® Scalable processor family includes many new technologies enabled and designed to improve and increase the networking workload performance.

The relevant encryption instructions include:

- VPMADD2 - vectorized multiplication and addition vector instruction
- vAES - vector version of the Intel® AES-NI instructions
- vCLMULQDQ - vectorized carry-less multiplication
- SHA-NI - secure hash algorithm - new instructions

The combination of vAES and vPCLMULQDQ on wide registers that are available on the Intel® AVX-512 architectures further speed-up AES modes such as AES-CTR and AES-CBC. VPMADD2 is targeted at significantly reducing the instructions needed to generate public/private keys as part of an RSA-2K sign operation. SHA-NI attempts to improve hashing functions used in cryptographic protocols such as Secure Sockets Layer (SSL)/Transport Layer Security (TLS) as well as helping with data deduplication in storage workloads.

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

- New Intel® AVX-512 instruction set support for accelerated processing of vectorized instructions https://www.intel.com/content/www/us/en/architecture-and-technology/avx-512-overview.html
- New Intel® Speed Select Technology (SST) power management technologies for increased power-aware performance https://www.intel.com/content/www/us/en/architecture-and-technology/speed-select-technology-article.html
- Intel® AVX-512 – Packet Processing with Intel® AVX-512 Instruction Set Solution Brief https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-packet-processing-with-intel-avx-512-instruction-set-solution-brief
- Intel® AVX-512 – Instruction Set for Packet Processing Technology Guide https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-instruction-set-for-packet-processing-technology-guide
- Intel® AVX-512 – Writing Packet Processing Software with Intel® AVX-512 Instruction Set Technology Guide https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-writing-packet-processing-software-with-intel-avx-512-instruction-set-technology-guide

2.2.2 Intel® Ethernet 810 Series Network Adapter

Intel® Ethernet Network Adapter E810-2CQDA2 PCIe 4.0 NIC is a dual 100 Gbps port network adapter designed to optimize networking workloads including NFV. Intel® Ethernet 810 Series adapters contain technologies such as:

- Intelligent Flow Direction: Receiver Side Scaling (RSS)
- Comprehensive Network Virtualization Overlay Protocols Support

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

- 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 Series, refer to:

- Ethernet Products: https://ark.intel.com/content/www/us/en/ark/products/series/184846/intel-ethernet-network-adapter-e810-series.html
- Intel® Ethernet Controller 800 Series – Dynamic Device Personalization (DDP) for Telecommunications Workloads Technology Guide: https://networkbuilders.intel.com/solutionslibrary/intel-ethernet-controller-800-series-device-personalization-ddp-for-telecommunications-workloads-technology-guide
- Intel® Ethernet Controller 700/800 Series – Dynamic Device Personalization Support for CNF with Kubernetes Technology Guide: https://networkbuilders.intel.com/solutionslibrary/intel-ethernet-controller-700-series-dynamic-device-personalization-support-for-cnf-with-kubernetes-technology-guide

2.2.3 Intel® Multi-Buffer Crypto for IPsec Library

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 this library, the user can make complete use of the latest cryptographic accelerations provided by Intel, including the new vAES and vPCLMULQDQ instructions. The Intel AVX-512 accelerated instructions allow processing of up to four 128-bit AES blocks in parallel, thereby, theoretically achieving four-times better performance than the previous generation Intel processor. Moreover, this library hides all implementation details to accommodate different CPU flags (SSE, AVX, AVX2, AVX-512) behind the APIs, to ensure highly optimized cryptographic operation results for all Intel® CPUs in the market and provides the user with seamless transition of their code into new 3rd Generation Intel® Xeon® Scalable processor based systems.

For more information about Intel® Multi-Buffer Crypto for IPsec library, Intel vAES and VPCLMULQDQ instructions, refer to:

- https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/fast-multi-buffer-ipsec-implementations-ia-processors-paper.pdf
- https://newsroom.intel.com/articles/crypto-acceleration-enabling-path-future-computing/#gs.n6t02b
- https://github.com/intel/intel-ipsec-mb

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

FD.io is a Linux Foundation open source project that provides fast network packets processing capability. 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, called “vector of packets” or “packet vector”. Each function block treats a packet vector (currently maximum 256 packets) as input and processes them in the same manner. This helps to maximize highly efficient utilization of CPU instruction cache (I-cache). In addition, VPP innovatively adopts a packet processing graph as its core design, where each function block is abstracted as a graph node. The graph nodes are organized in a tree-shaped graph by registering the “next” output nodes initially or during runtime. The packet vectors flow from NIC RX nodes all the way to TX nodes (or dropped) based on the processed destinations in each graph node within.

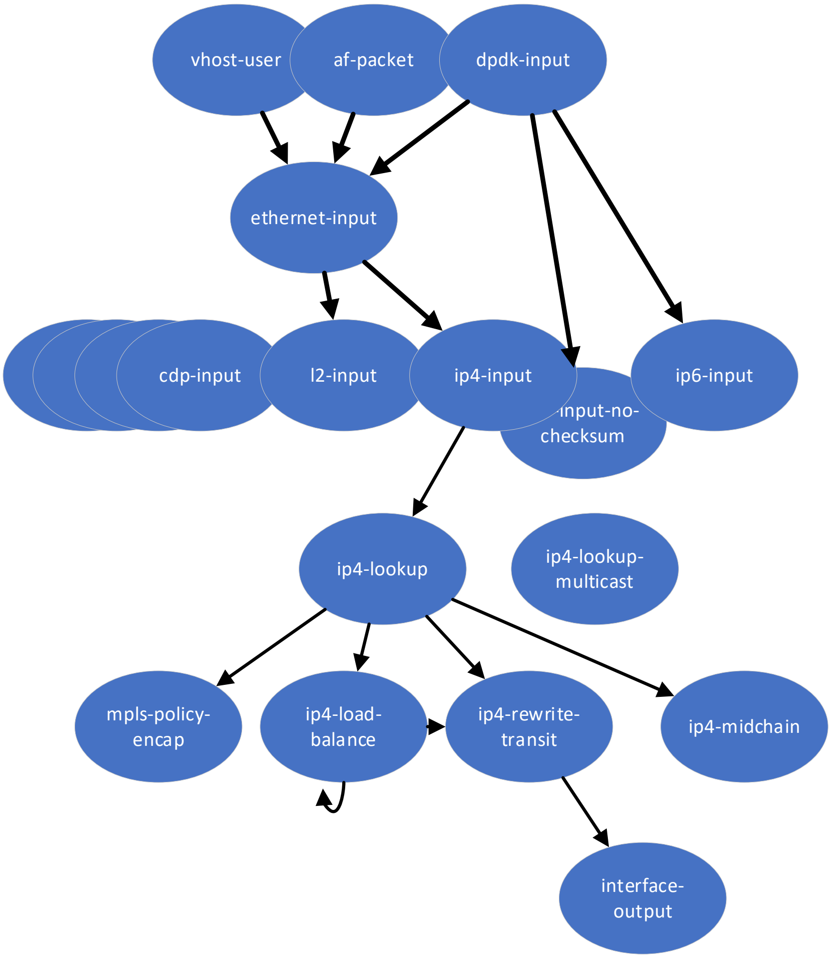


Figure 1. VPP Packet Processing Graph

The packet processing graph has the distinctive advantage of extreme flexibility, i.e., the new graph nodes can be "plugged in" anywhere and existing ones can be bypassed via simple software or real-time command line configuration. Moreover, the connected graph nodes become the packet processing pipeline. The packets received by the NIC receive port are processed by one graph node after another before being sent to the NIC transmit port. This mode of working is called "run-to-completion". The run-to-completion design of VPP make the packets buffer prefetching into the CPU's data cache predictable and efficient, while maximizing the CPU's instruction utilization in the meantime. This makes VPP very efficient.

For more information on VPP, refer to:

- https://wiki.fd.io/view/VPP
- https://fd.io/docs/vpp/master/usecases/contiv/vpp_config
- https://wiki.fd.io/view/VPP/Command-line_Interface_(CLI)_Guide

VPP IPsec

VPP IPsec is an important component in VPP to provide secure, reliable, and fast networking applications. VPP IPsec provides a set of easy-to-use CLI and VAPI commands for user to configure 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 largest resource-consuming procedure within IPsec is the cryptographic operation. To ensure both the performance and the flexibility of cryptographic operation, VPP IPsec takes advantage of crypto infrastructure.

VPP Crypto Infrastructure and Engines

The VPP crypto infrastructure is a framework that supports different crypto engines working as plugins to performance symmetric crypto operations. So far there are three crypto engines:

- **Native engine:** The crypto engine that is specifically designed for VPP achieves fastest crypto processing efficiency with limited algorithms supported. vAES and vPCLMULQDQ acceleration of AES encryption/decryption are automatically enabled if the application is running on the latest architecture CPUs.
- **IPsecMB engine:** Integration layer to Intel® Multi-Buffer Crypto for IPsec Library with extended crypto algorithm support list with less performance compared to the native engine. vAES and vPCLMULQDQ acceleration of AES encryption/decryption are automatically enabled if the application is running on the latest 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 allows the most performant crypto implementation to be used for a specific algorithm.

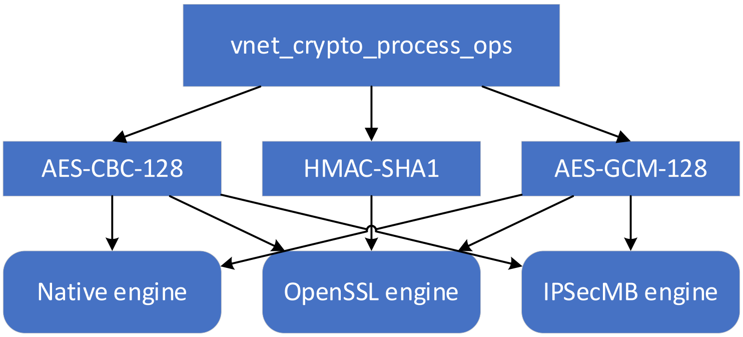


Figure 2. VPP Crypto Infrastructure

For more information about VPP crypto Infrastructure and VPP IPsec, refer to https://wiki.fd.io/view/VPP/IPSec_and_IKEv2.

3 Deployment

The Device Under Test (DUT) consists of Supermicro* X12DPG-QT6 Dual Socket computer system. In each socket, a 3rd Generation Intel® Xeon® Scalable processor and three Intel® Ethernet Network Adapter E810-2CQDA2 are installed. Each Intel® E810 adapter contains two 100 Gbps ports and takes 1 PCIe Gen-4 x 16 socket in the Supermicro board. The DUT is connected to an Ixia* platform (a hardware test and simulation platform) to generate packet traffic to the DUT ports and determine the throughput at the tester site. The Ixia is used to implement RFC2544 on the DUT.

Table 3 displays the full system setup[4].

Table 3. System Setup

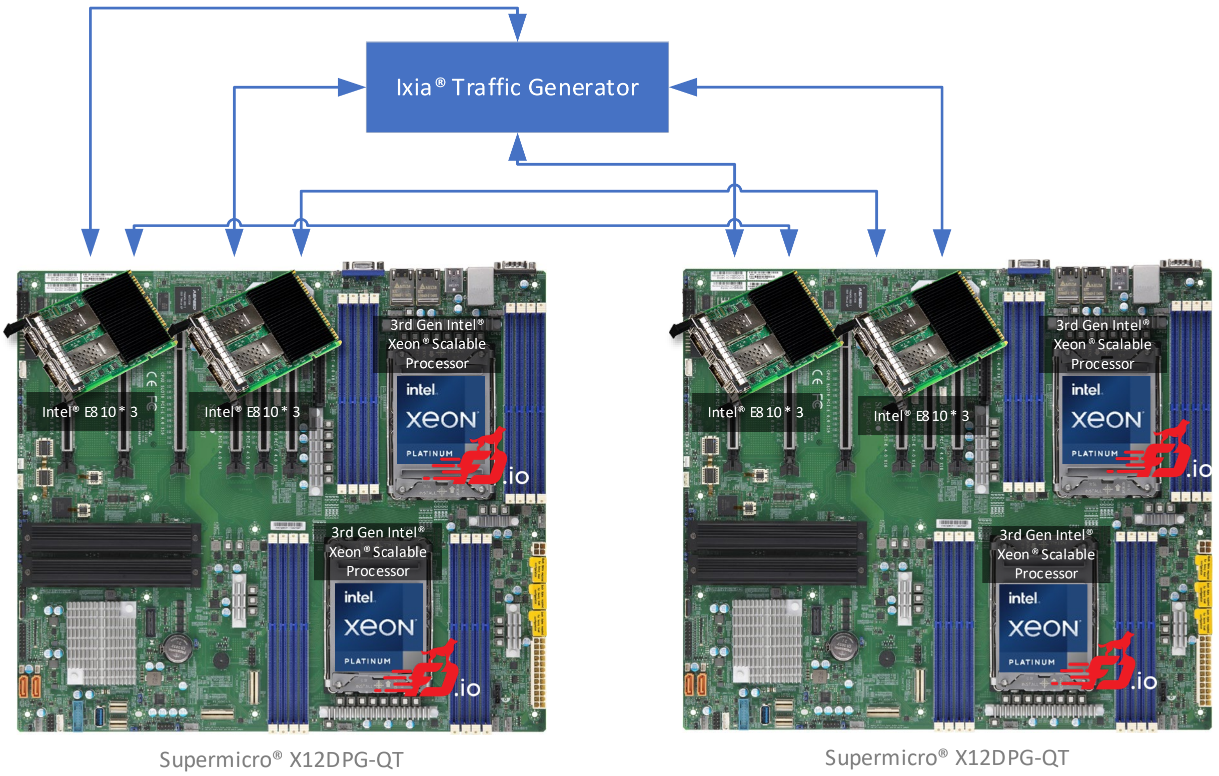
Item	Description
Server Platform	Supermicro X12DPG-QT6
CPU	3rd Generation Intel® Xeon® Scalable processor 8360Y @2.4GHz 36 CPU cores x 2 NUMA nodes
Memory	256GB: 16GB x 8 DIMMs x 2 NUMA modes @ 3200MHz DDR4
NIC	Intel® Ethernet Network Adapter E810-2CQDA2 x 6
NIC Firmware Version	2.30 0x80005d1c
BIOS	Version 1.0, build date 03/24/2021. Default BIOS configuration. Turbo boosting is turned OFF. Full BIOS settings can be found in Appendix A.1.

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

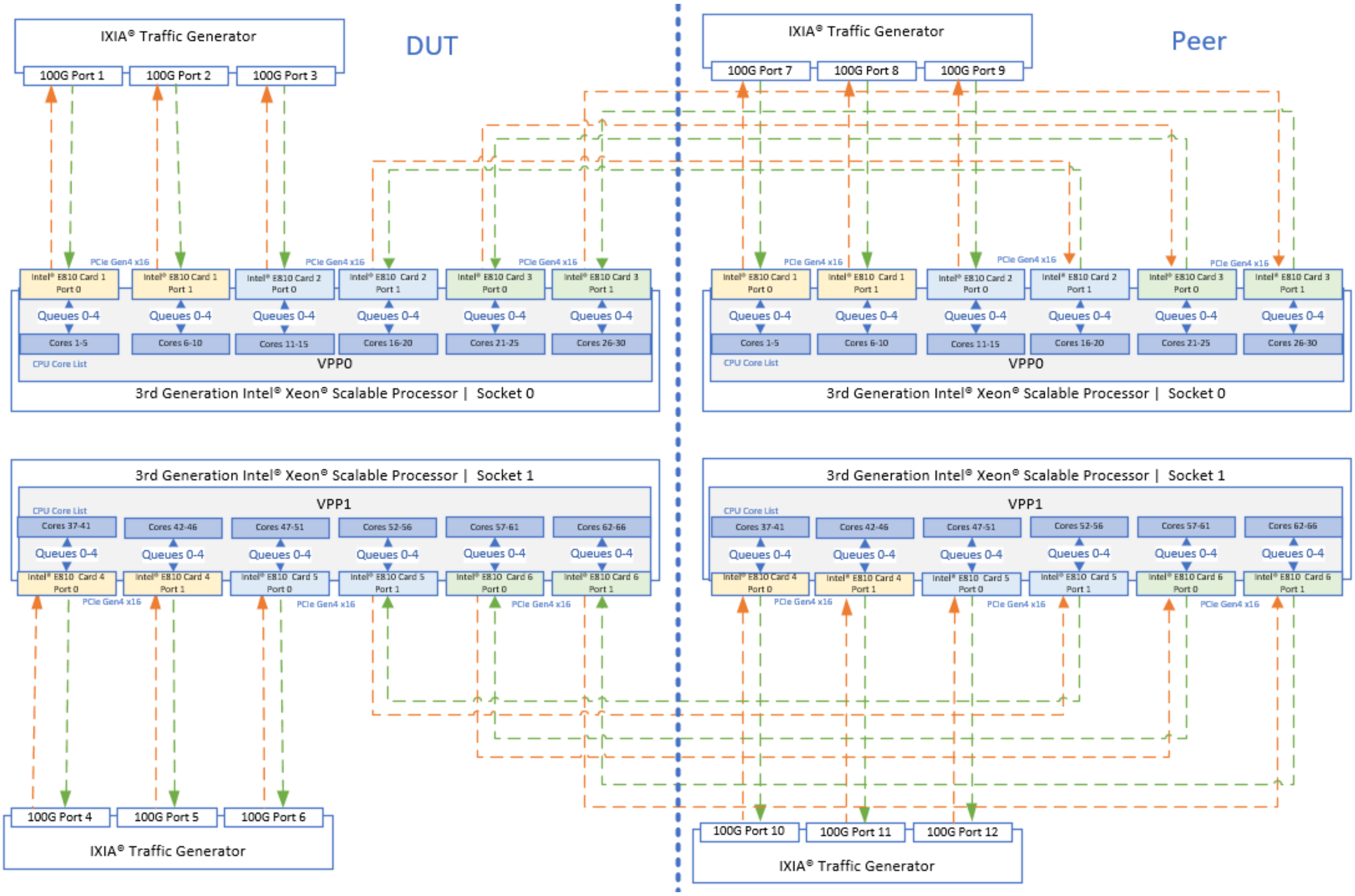
Microcode	0x0xd000270
Operating System	Ubuntu 20.04 LTS
Linux Kernel Version	5.4.0-40-generic
Kernel GRUB command	hugepagesz=1G hugepages=8 isolcpus=1-35,73-107,37-71,109-143 default_hugepagesz=1G rcu_nocbs=1-35,73-107,37-71,109-143 nohz_full=1-35,73-107,37-71,109-143 panic=30 init=/sbin/init net.ifnames=0 nmi_watchdog=0 audit=0 nosoftlockup hpet=disable mce=off tsc=reliable numa_balancing=disable memory_corruption_check=0 workqueue.power_efficient=false
VPP Version	21.01

3.1 Deployment Setup

Figure 3 below shows the deployment setup of DUT. To perform a complete IPsec outbound and inbound protection test, we deployed two DUTs each with two Xeon® Scalable processors to handle IPsec outbound and inbound protections to the packets sent by Ixia.


Ixia® Traffic Generator
Intel® E810 * 3
Intel® E810 * 3
3rd Gen Intel® Xeon® Scalable Processor
3rd Gen Intel® Xeon® Scalable Processor
Supermicro® X12DPG-QT
Intel® E810 * 3
Intel® E810 * 3
3rd Gen Intel® Xeon® Scalable Processor
3rd Gen Intel® Xeon® Scalable Processor
Supermicro® X12DPG-QT

A: Brief Deployment Diagram



B: Detailed Deployment Diagram

Figure 3. Deployment Diagram

3.1.1 DUT NIC Ports and Ixia Ports Connection

This test uses a total of 12 100 Gbps Ixia ports to generate and receive up to 1.2 Tbps traffic. Every 3rd Generation Intel® Xeon® Scalable processor socket is installed with three Intel® E810 adapters, six 100Gbps ports. Three of the 100 Gbps ports connect to three Ixia ports, and the rest connect to three E810 ports of the peer DUT. For example, from Figure 3, we can see Ixia port 1 to 3 are connected to DUT 0's E810 adapter 1's port 0 and 1, and adapter 2's port 0. The adapter 2's port 1 and adapter 3's port 0 and 1 are connected to DUT 1's Intel® E810 adapter 2's port 1 and adapter 3's port 0 and 1, respectively.

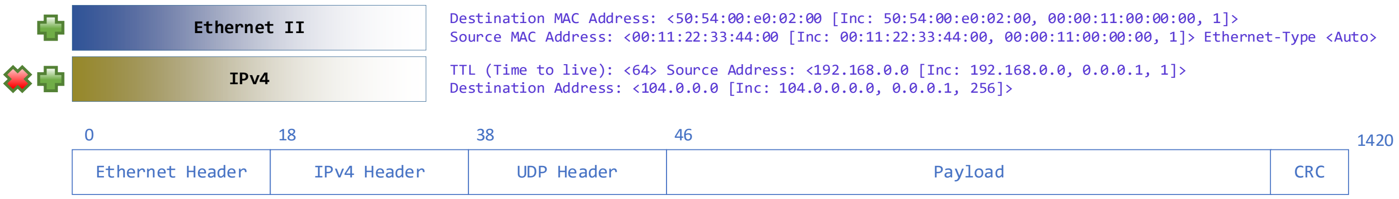
3.1.2 NIC Queue Configuration and CPU Utilization

Every Intel® E810 adapter port is configured with four NIC receive queues and transmit queues (Queues 0-4 in the figure) to share with four CPU cores respectively. We used 6 * 5 = 30 CPU cores in each socket to handle IPsec outbound and inbound processing. In total, we used 60 dedicated CPU cores for the test in each DUT, with no hyper-threading.

3.1.3 Ixia Flow Configuration

Ixia port 1 to 12 sends plain IPv4 packets, shown as green arrows in Figure 3, to the NIC ports connected. Every Ixia port sends the packets with unique 256 flows different from other Ixia ports. Each flow contains unique destination MAC and IP addresses. It is expected that each Intel® E810 adapter port maps a flow to a specific NIC queue and evenly distribute all packets across the four NIC receive queues through the Receive Side Scaling (RSS) technology in the NIC. In the end, each receive queue should have fixed 256 / 5 = 51.2 flows and the Ixia should send evenly distributed 256 * 12 = 3072 flows packets at any specific time.

A sample Ixia port configuration can be found in Figure 4. As mentioned, every port has different source and destination MAC IP addresses, hence a total of 12 different Ixia port configurations are set.



Ethernet II

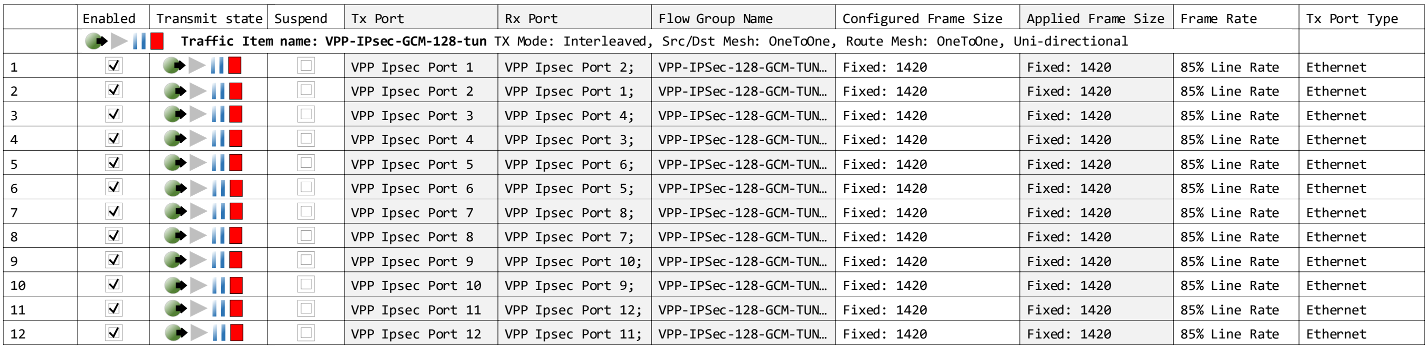
Destination MAC Address: <50:54:00:e0:02:00 [Inc: 50:54:00:e0:02:00, 00:00:11:00:00:00, 1]>
Source MAC Address: <00:11:22:33:44:00 [Inc: 00:11:22:33:44:00, 00:00:11:00:00:00, 1]> Ethernet-Type <Auto>

IPv4

TTL (Time to live): <64> Source Address: <192.168.0.0 [Inc: 192.168.0.0, 0.0.0.1, 1]>
Destination Address: <104.0.0.0 [Inc: 104.0.0.0.0, 0.0.0.1, 256]>

Figure 4. A Sample Ixia Port Packet Configuration

Figure 5 illustrates the configuration flow of all 12 Ixia ports. In the figure, we can see every port is transmitting fixed 1420 byte packets. The maximum throughput of an Ixia port is 100 Gbps[5].



	Enabled	Transmit state	Suspend	Tx Port	Rx Port	Flow Group Name	Configured Frame Size	Applied Frame Size	Frame Rate	Tx Port Type
	▶ ▶ ‖ ■ **Traffic Item name: VPP-IPsec-GCM-128-tun** TX Mode: Interleaved, Src/Dst Mesh: OneToOne, Route Mesh: OneToOne, Uni-directional									
1	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 1	VPP Ipsec Port 2;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
2	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 2	VPP Ipsec Port 1;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
3	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 3	VPP Ipsec Port 4;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
4	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 4	VPP Ipsec Port 3;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
5	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 5	VPP Ipsec Port 6;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
6	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 6	VPP Ipsec Port 5;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
7	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 7	VPP Ipsec Port 8;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
8	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 8	VPP Ipsec Port 7;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
9	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 9	VPP Ipsec Port 10;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
10	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 10	VPP Ipsec Port 9;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
11	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 11	VPP Ipsec Port 12;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet
12	☑	▶ ▶ ‖ ■	□	VPP Ipsec Port 12	VPP Ipsec Port 11;	VPP-IPSec-128-GCM-TUN…	Fixed: 1420	Fixed: 1420	85% Line Rate	Ethernet

Figure 5. Screenshot of Ixia Network Application Flow Group Configuration

3.2 VPP Application Configuration

Within each DUT, we ran two VPP processes simultaneously (noted as “VPP0” and “VPP1” in Figure 3), one on each DUT socket. In total, there are four VPP processes running on two DUTs. In this section, we describe the VPP configurations and CLI commands used in the test.

3.2.1 VPP Startup.conf

A VPP process requires a dedicated “startup.conf” file that contains specific configurations for the VPP application to run as desired. Since there are four VPP processes running in parallel, we need to configure four startup.conf files, respectively. Figure 6 shows a sample startup.conf for the VPP0 process. The content of the other startup.conf file is similar to that in Figure 6 but with different “corelist-workers” and DPDK port configuration section.

```
cpu {
        # CPU core utilization for DUT socket 0
        main-core 0
        corelist-workers 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,25,26,27,28
}
buffers { buffers-per-numa 133320 }
dpdk {
        socket-mem 2048
        no-tx-checksum-offload
        dev default{
                num-tx-desc 512
                num-rx-desc 512
        }
```

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

```
    # Intel® E810 port utilization for DUT0 socket 0, 6 ports in total.
    # Only one port is showing here, the rest 5 ports are omitted.

    dev 0000:17:00.0

    {

        workers 0,1,2,3

    }

    no-multi-seg

}

memory { main-heap-size 2G }

statseg { size 1G }
```

Figure 6. VPP Startup.conf

3.2.2 VPP CLI commands

As shown in Figure 3, a VPP process receives plain IPv4 packets from three Ixia ports and IPsec ESP packets from three peer DUT ports. In other words, it handles both IPsec outbound and inbound processes. After a packet is processed it needs to forward the encrypted packets to the ports connected to peer DUT ports or forward the decrypted packets to the ports connected to the Ixia ports.

To achieve this goal, in addition to startup.conf file, each VPP process requires CLI command input to understand how to deal with the packets received. Our test attempts to let both VPP processes perform the same following steps:

- Perform polling of every NIC's receive queue.
- When a packet is received, check if the packet matches the flow rule desired.
- If the packet matches the IPsec outbound rule, perform IPsec IPv4 tunnel outbound operation with AES-GCM-128 encryption.
- If the packet matches the IPsec inbound rule, perform IPsec IPv4 tunnel inbound operation with AES-GCM-128 decryption.
- Send the encrypted or decrypted packets through the port that matches the TX forwarding rules.

Hence, we used the VPP CLI commands shown in Figure 7. The commands are divided into two sections:

- The commands used to configure NIC ports: Figure 7 contains the configuration for only one NIC, but in the test all 12 NIC ports (six ports in VPP0 and six ports in VPP1) must be configured with different MAC and IP addresses.
- The commands used to create IPsec SA/ESP rules and routing rules for plain and encrypted packets: We have listed the commands to create only one IPsec SA/ESP rule. However, to cover all flows in the test, all 6 ports * 256 flows/port * 2 VPP processes = 3072 flows must be configured with different IP addresses.

```
# Configure the NIC port, needed to be set for all 6 ports

set interface state HundredGigabitEthernet17/0/0 up

set interface mtu 2024 HundredGigabitEthernet17/0/0

set interface MAC address HundredGigabitEthernet17/0/0 00:22:33:44:55:0

set interface ip address HundredGigabitEthernet17/0/0 255.0.0.128/8

set int promiscuous on HundredGigabitEthernet17/0/0


# The following commands need to be set for all 6 * 256 = 1536 flows.

# Create a SW tunnel interface and 2 IPsec Security Associations (SAs). For any packets going into this
interface an IPsec ESP AES-GCM-128 tunnel operation will be performed.

create IPsec tunnel local-ip 255.0.0.128 remote-ip 255.0.0.129 local-spi 255128 remote-spi 255129 local-
crypto-key 2b7e151628aed2a6abf7158809cf4f3d remote-crypto-key 2b7e151628aed2a6abf7158809cf4f3d crypto-alg
aes-gcm-128  integ-alg none salt 0x12345678

# Configure the rule to route plain IPv4 packets with destination address "104.0.0.0" to the SW interface
just created.
```

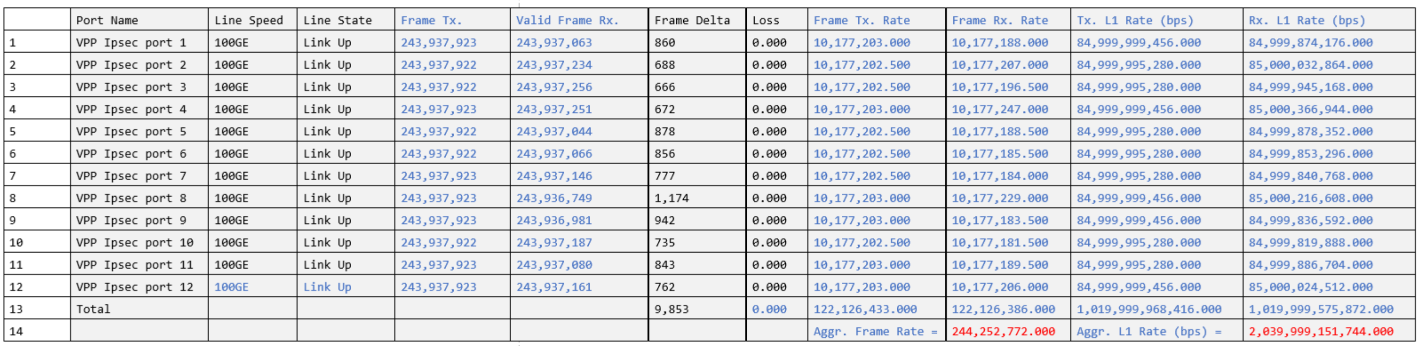
```
ip route add 104.0.0.0/32 via ipip0
set int state ipip0 up
# Let the SW interface owning same IP address as the HW interface.
set int unnum ipip0 use HundredGigabitEthernet17/0/0
# Create a new ARP rule for encrypted packets with new tunnel header.
set ip neighbor HundredGigabitEthernet17/0/0 255.0.0.129 3d:3f:12:11:00:11
# Configure the rule to route decrypted IPv4 packets with destination address "204.0.0.0" to the NIC
port.
ip route add 204.0.0.0/32 via HundredGigabitEthernet18/0/0 255.0.1.129 3d:3f:12:11:00:aa
```

Figure 7. VPP CLI Commands for One VPP Process

4 Results

After the system setup is complete and VPP applications are up and running with all commands injected, we started the Ixia traffic generator to transmit the flows defined in Section 3.1.3. We used RFC 2544 zero packet loss test with accepted packet loss rate of 0%. The test was set to run for 60 seconds and the benchmark results were collected at 48th second. The final throughput results are shown in Figure 8[6] below.

Note: These test results are as of March 30, 2021.



	Port Name	Line Speed	Line State	Frame Tx.	Valid Frame Rx.	Frame Delta	Loss	Frame Tx. Rate	Frame Rx. Rate	Tx. L1 Rate (bps)	Rx. L1 Rate (bps)
1	VPP Ipsec port 1	100GE	Link Up	243,937,923	243,937,063	860	0.000	10,177,203.000	10,177,188.000	84,999,999,456.000	84,999,874,176.000
2	VPP Ipsec port 2	100GE	Link Up	243,937,922	243,937,234	688	0.000	10,177,202.500	10,177,207.000	84,999,995,280.000	85,000,032,864.000
3	VPP Ipsec port 3	100GE	Link Up	243,937,922	243,937,256	666	0.000	10,177,202.500	10,177,196.500	84,999,995,280.000	84,999,945,168.000
4	VPP Ipsec port 4	100GE	Link Up	243,937,923	243,937,251	672	0.000	10,177,203.000	10,177,247.000	84,999,999,456.000	85,000,366,944.000
5	VPP Ipsec port 5	100GE	Link Up	243,937,922	243,937,044	878	0.000	10,177,202.500	10,177,188.500	84,999,995,280.000	84,999,878,352.000
6	VPP Ipsec port 6	100GE	Link Up	243,937,922	243,937,066	856	0.000	10,177,202.500	10,177,185.500	84,999,995,280.000	84,999,853,296.000
7	VPP Ipsec port 7	100GE	Link Up	243,937,923	243,937,146	777	0.000	10,177,202.500	10,177,184.000	84,999,995,280.000	84,999,840,768.000
8	VPP Ipsec port 8	100GE	Link Up	243,937,923	243,936,749	1,174	0.000	10,177,203.000	10,177,229.000	84,999,999,456.000	85,000,216,608.000
9	VPP Ipsec port 9	100GE	Link Up	243,937,923	243,936,981	942	0.000	10,177,203.000	10,177,183.500	84,999,999,456.000	84,999,836,592.000
10	VPP Ipsec port 10	100GE	Link Up	243,937,922	243,937,187	735	0.000	10,177,202.500	10,177,181.500	84,999,995,280.000	84,999,819,888.000
11	VPP Ipsec port 11	100GE	Link Up	243,937,923	243,937,080	843	0.000	10,177,203.000	10,177,189.500	84,999,995,280.000	84,999,886,704.000
12	VPP Ipsec port 12	100GE	Link Up	243,937,923	243,937,161	762	0.000	10,177,203.000	10,177,206.000	84,999,999,456.000	85,000,024,512.000
13	Total					9,853	0.000	122,126,433.000	122,126,386.000	1,019,999,968,416.000	1,019,999,575,872.000
14								Aggr. Frame Rate =	244,252,772.000	Aggr. L1 Rate (bps) =	2,039,999,151,744.000

Figure 8. Screenshot of Ixia Networks Application Throughput Statistics

As shown in the figure, the Ixia ports collected the aggregated packet receive rate of 244.252 Mpps, or 2.040 Tbps. This is the accumulation results from two DUTs. The VPP applications running on the latest Intel® Xeon® Scalable Processor achieved a total of 1.012 Tbps and 122.126 Mpps IPsec AES-GCM-128 tunnel protection throughput with 48 CPU cores. On average, every CPU core processed 2.544 Mpps/21.081Gbps data. Table 4 below shows the statistics result in detail.

Table 4. Statistics Result

Item	Description
Number of CPU Cores Utilized	24 per numa node, 48 per system
IPsec Inbound and Outbound Throughput Rate Per CPU Core	2.544 Mpps, 21.081 Gbps
Aggregated IPsec Throughput Per System	122.126 Mpps, 1.012 Tbps
Aggregated IPsec Throughput in Total	244.252 Mpps, 2.040 Tbps

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

5 Summary

This guide demonstrates how the Intel® AVX-512 and Intel® AES-NI instructions provided in the latest Intel® Xeon® Scalable processor, and enabled in FD.io VPP through the Intel® Multi-Buffer Crypto for IPsec library, can be used to achieve over 1 Tbps of No Drop Rate IPsec AES-GCM-128 tunnel encryption throughput[7].

It also describes the underlying technologies, the challenges faced, and the hardware and software configurations used to achieve this 1 Tb figure. While these hardware and software configurations are specific to this setup, the technologies enabled by these hardware and software configurations are intended to be used as a reference for anyone looking to optimize their IPsec throughput.

End users are encouraged to test these latest technologies for themselves to evaluate the performance improvements available in the AVX-512 and AES-NI instructions provided in the latest Intel® Xeon® Scalable processor.

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

Appendix A

A.1 System BIOS Settings[8]

Menu (Advanced)	Path to BIOS setting	BIOS Function	BIOS Setting
CPU Configuration	CPU Configuration	Hyper-threading [ALL]	Enabled
Power Configuration	CPU Configuration -> Advanced Power Management -> CPU P State Control	EIST PSD Function	Enabled
		Turbo Mode	Disabled
		Activate PBF	Disabled
		Configure PBF	Disabled
		Speedstep (P-States)	Enabled
	CPU Configuration -> Advanced Power Management -> Hardware PM State Control	Hardware P-States	Native Mode
	CPU Configuration -> Advanced Power Management -> CPU C State Control	Autonomous Core C State	Enabled
		CPU C6 Report	Enabled
		Enhanced Halt State (C1E)	Enabled
	CPU Configuration -> Advanced Power Management -> Package C State Control	Package C State	Auto
	CPU Configuration -> Advanced Power Management	Power Performance Tuning	OS Controls EPB
		Energy Perf Bias CFG Mode	Balanced Performance
		Power Technology	Custom
IIO Configuration	Chipset Configuration -> North Bridge -> IIO Configuration -> IOAT Configuration	Relaxed Ordering	No
	Chipset Configuration -> North Bridge -> IIO Configuration -> Intel VT for Directed I/O (VT-D)	Intel VT for Directed I/O (VT-D)	Disabled
	Chipset Configuration -> North Bridge -> IIO Configuration -> CPU 1 Configuration	IOU0 (IIO PCIE BR1)	x8x8
		IOU1 (IIO PCIE BR2)	x8x8
		IOU2 (IIO PCIE BR3)	x8x8
	Chipset Configuration -> North Bridge -> IIO Configuration -> CPU 2 Configuration	IOU0 (IIO PCIE BR1)	x8x8
		IOU1 (IIO PCIE BR2)	x8x8
		IOU2 (IIO PCIE BR3)	x8x8
		Link L0P Enable	Disable

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

UPI Configuration	Chipset Configuration -> North Bridge -> UPI Configuration	Link L1P Enable	Disable
Memory Configuration	Chipset Configuration -> North Bridge -> Memory Configuration	Enforce POR	POR
USB Configuration	Chipset Configuration -> South Bridge	XHCI Hand-Off	Enabled

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

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

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.

Your costs and results may vary.

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