Technology Guide

Intel® QAT - Accelerate WireGuard* Processing with 4th Gen Intel® Xeon® Scalable Processor

Authors

Fan Zhang

Georgii Tkachuk

Pablo De Lara Guarch

Tomasz Kantecki

1 Introduction

WireGuard*, as a new and fast-growing network security protocol, has the characteristics of simplicity, usefulness, and security. The industry is welcoming WireGuard as the next generation VPN protocol, in addition to IPsec and TLS protocols. Currently, WireGuard has been officially integrated in Linux, Windows, and FreeBSD Kernel, and its popularity is growing in production use. Early usage of WireGuard includes multi-cloud connectivity to secure nodes both in the same location as well as across locations as illustrated in Figure 1.

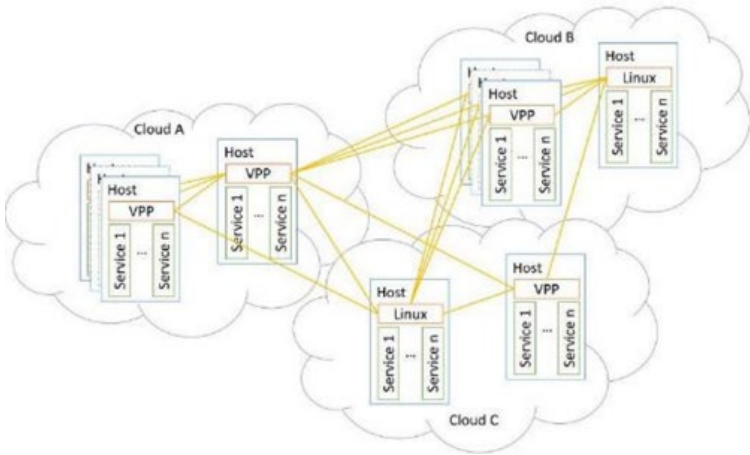

Figure 1. WireGuard securing nodes within and across Cloud locations

There are many open-source projects that implement WireGuard, mainly implemented in C, Golang, and Rust. WireGuard ecosystem is strong and easy to use. A high-performance WireGuard can be assisted by the latest Intel silicon technology 4th Gen Intel® Xeon® Scalable processor (code named Sapphire Rapids[1]). Intel has two features to accelerate a crypto processing of WireGuard: Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instruction and Intel® QuickAssist® Technologies (Intel® QAT).

This document highlights the performance difference of WireGuard processed by VPP WireGuard with Intel® AVX-512 instructions, and VPP WireGuard with Intel® QAT on the same 4th Gen Intel Xeon Scalable processor.

This document is intended for WireGuard developers, or anyone looking for a better and faster WireGuard solution to accelerate the existing network infrastructure. The technologies enabled here can be used as a reference point for improving performance in any WireGuard or networking deployment.

This document is part of the Network Transformation Experience Kits.

[1] 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.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	January 2023	Initial release.

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
AES-NI	Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI)
FD.io	Fast Data input/output
IPSec	Internet Protocol Security
IPsec-MB	Intel® Multi-Buffer Crypto for IPsec Library
TLS	Transport Layer Security
VAES	Vectorized Advanced Encryption Standard
VPP	Vector Packet Processing
VPN	Virtual Private Network
WireGuard	Encrypted Virtual Private Network (VPN) Communication Protocol

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
Intel® QuickAssist® Technology (Intel® QAT)	https://www.intel.com/content/www/us/en/architecture-and-technology/intel-quick-assist-technology-overview.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 reference guide to show the performance difference of different WireGuard implementations running on the same 4th Gen Intel Xeon Scalable processor. The document is also intended to showcase the impressive 204 Gbps throughput that can be achieved on the latest Intel® Xeon® CPU. The hardware optimization of the CPU is achieved through both the Intel® AVX-512 instructions utilization, and up to 200 Gbps symmetric crypto encryption/decryption Intel QuickAssist Technology (Intel QAT) Gen 4. The software optimization is achieved through the Fast Data input/output (FD.io) Vector Packet Processing (VPP) open-source packet-processing stack, which for WireGuard cryptography, employs the new Intel AVX-512 instructions for Chacha20-Poly1305 AEAD cryptographic encryption/decryption, powered by latest Intel® IPsec Multi-Buffer (intel-ipsec-mb) library.

In this document, we will describe how the test system was set up and deployed using the technologies listed above, and how it was used to achieve over 8 Gigabits per second bidirectional single WireGuard peer Encryption and Decryption throughput with software, or 46 Gigabits per second bidirectional single WireGuard peer Encryption and Decryption throughput with Intel QAT hardware accelerated VPP WireGuard implementation.

While this guide goes into detail on how this 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 WireGuard performance.

2.1 Challenges Addressed

The industry is welcoming WireGuard as the next generation VPN protocol. Currently, WireGuard has been officially integrated in Linux, Windows, and FreeBSD Kernel, and its implementations written in C, Golang, and Rust have reached tens and the number is growing. However, a performant user-space WireGuard implementation is still lacking.

In this article we are describing VPP WireGuard Implementation that takes advantage of both Intel® AVX-512 instructions and Intel® QAT hardware Chacha20-Poly1305 cryptographic operation offloading and highlight the WireGuard performance achieved in the 4th Gen Intel® Xeon® Scalable Processor.

2.2 Technology Description

2.2.1 4th Gen Intel® Xeon® Scalable Processor

The new 4th Gen Intel® Xeon® Scalable processor, based on Intel 7 processor node, is a revolutionary computer platform designed for workload acceleration. It has maximum 56 CPU cores, which is 40% more than the 3rd Gen Intel® Xeon® Scalable processor - not to mention each CPU core has 19% more IPC over previous generation 3rd Gen Intel Xeon Scalable processor. It offers 8-Channel 4800MT/s DDR5 memory, which is more than 1.5 times memory bandwidth of the 3rd Gen Intel Xeon Scalable processor; moreover, its 80 PCIe 5.0 lanes provide more than 100% extra PCIe bandwidth over the last gen processors.

The 4th Gen Intel Xeon Scalable processor also inherits the Intel AVX-512 instruction accelerations. The relevant encryption instructions to boost cryptographic operation performance, such as VPMADD2 (vector instruction that does integer multiply accumulate), vAES (vector version of the AES-NI instructions), vCLMUL(vectorized carryless multiple), and SHA-NI (secure hash algorithm new instructions) now are further enhanced by the 19% Instruction Per Cycle (IPC) performance improvement over the 3rd Gen Intel Xeon Scalable processor.

The 4th Gen Intel Xeon Scalable processor has the built-in Intel QAT (4th Gen) that provides up to 200 Gbps symmetric crypto processing capability. The cryptography operation offloading feature provided by Intel® QAT helps free up the CPU cycles from cryptography so that the CPU can spend cycles to process other workloads.

Refer to the following references 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:
  https://www.intel.com/content/www/us/en/architecture-and-technology/avx-512-overview.html

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

  https://www.intel.com/content/www/us/en/architecture-and-technology/speed-select-technology-article.html

2.2.2 Intel® QuickAssist Technology Gen 4

Intel QuickAssist Technology provides hardware acceleration to assist with the performance demands of securing and routing Internet traffic and other workloads, such as compression and wireless 5G and 4G LTE algorithm offload, thereby reserving processor cycles for application and control processing. The 4th Gen Intel Xeon Scalable processor contains the built-in Intel® QAT accelerator that assists processing up to 200Gbps symmetric crypto workload offloading, such as Chacha20-Poly1305 AEAD algorithm processing. As a result, the CPU can be freed to process the rest of the WireGuard stack as well as the remaining application needs.

For more information on Intel® QAT, refer to:
https://www.intel.com/content/www/us/en/architecture-and-technology/intel-quick-assist-technology-overview.html

2.2.3 Intel® Ethernet 800 Series Network Adapter

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

- 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 on Intel® Ethernet Network Adapter E810-2CQDA2, refer to:
https://ark.intel.com/content/www/us/en/ark/products/210969/intel-ethernet-network-adapter-e8102cqda2.html

[2] Requires x16 slot bifurcation

2.2.4 Intel® Multi-Buffer Crypto for IPsec Library

Intel® Multi-Buffer Crypto for IPsec Library is a family of highly optimized software implementations of symmetric cryptographic algorithms. With the rich and easy-to-use APIs provided by Intel® IPsec-MB library, the user can easily make full use of the CPU latest cryptographic accelerations provided by Intel including the new Intel AVX512 vector instructions. Intel-ipsec-mb library provides an optimized implementation of Chacha20-Poly1305, using Intel AVX512 vector instructions to compute the ciphertext/plaintext and digest of the buffer. On the ciphering side, up to sixteen 64-byte Chacha20 blocks are computed in parallel, encrypting/decrypting up to 1KB of data in each iteration. On the authentication side, the data is digested parallelizing the computation of up to sixteen 16-byte blocks. To achieve this:

- Up to 16 powers of the Poly1305 key are precomputed, to be multiplied with the data blocks
- Multiplication is carried out with IFMA instructions and full reduction is performed only at the end of the message, to get the digest

For more information on Intel® IPsec-MB library, refer to:

- https://github.com/intel/intel-ipsec-mb
- https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/fast-multi-buffer-ipsec-implementations-ia-processors-paper.pdf

2.2.5 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 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.

VPP processes the packet in the burst manner, grouping up to 256 packets into a packet vector. To maximize the utilization of the CPU instruction cache (I-cache), VPP adopts the Packet Processing Graph as its core design. The graph nodes are organized as tree shape graph in VPP. The packet vectors will be flowing from NIC RX nodes all the way to TX nodes (or dropped) based on the processed destinations in each graph node within.

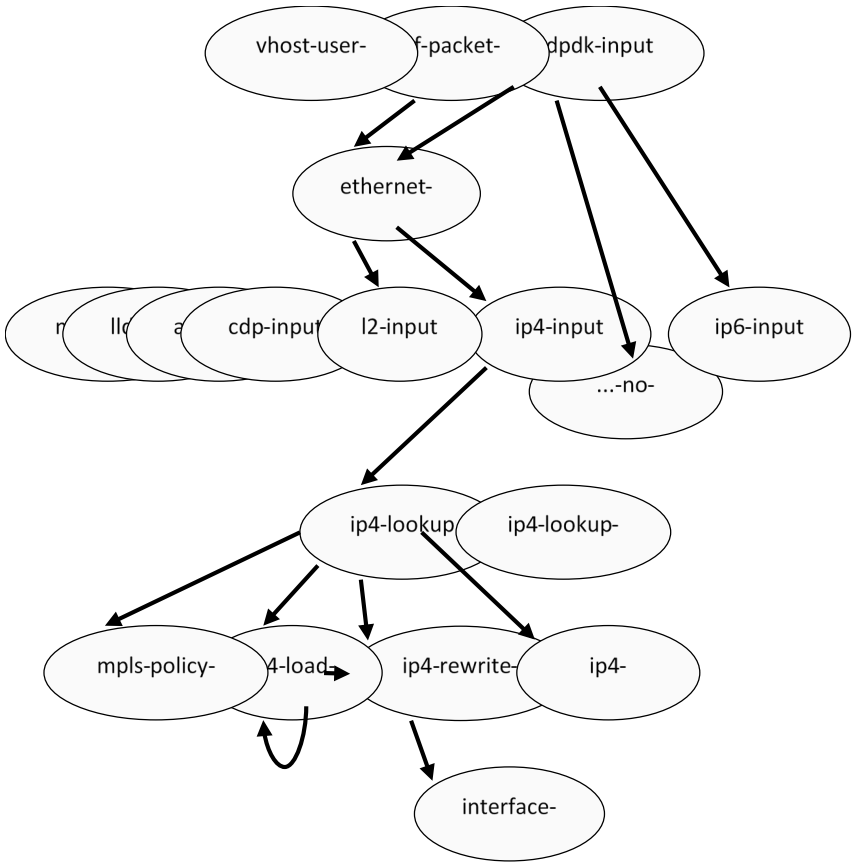

Figure 2. VPP packet processing graph

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

2.2.6 VPP WireGuard

VPP WireGuard is a new component in VPP to provide full WireGuard stack processing to VPP. VPP WireGuard provides a set of easy-to-use CLI and VAPI commands for user to configure WireGuard interfaces and peers in addition to existing rich routing and packet RX/TX functionalities, can perform handshake, rekeying, and WireGuard stack processing such as packet encapsulation, decapsulation, and cryptographic operations.

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

2.2.7 VPP Crypto Infrastructure and Engines

The VPP crypto infrastructure is a crypto framework that supports different crypto engines working as plugins to perform symmetric crypto operations. The following three crypto engines are included:

- **Built-in engine**: The crypto engine specifically designed for VPP that achieves high crypto processing efficiency but with limited algorithms supported. Some Intel AVX512 accelerations such as vAES and vPCLMUL are automatically enabled if the application is running on the latest CPU architecture.
- **IPsec MB engine**: Integration layer to Intel Multi-Buffer Crypto for IPsec Library with extended crypto algorithm. Intel AVX-512 acceleration of AES encryption/decryption is 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.

To maximize the crypto accelerators' performance, VPP crypto infrastructure also supports an asynchronous working mode. The workload crypto request is enqueued to the Intel® QAT and immediately continues to process the next packets instead of waiting for the current workload to complete processing. A function block running in a polling manner queries if previously enqueued workload is processed. This asynchronous working mode helps maximize Intel QAT and CPU packet processing efficiency. DPDK Cryptodev engine is one of the supported VPP crypto engines, which works in this mode and Intel QAT is controlled through it.

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 will be invoked to process the crypto operation. This flexible operation mode allows the most performant crypto implementation to be used for a specific algorithm.

3 Deployment

Table 3. System Setup

Item	Description
Server Platform	Pre-production Intel® reference platform (code name Archer City)
CPU	2 Sockets Pre-production Intel® Xeon® Platinum 8470N @1.70GHz
Memory	Manufacturer: Hynix*, Speed: 4800 MT/s, Number: 8 per socket, 1 DIMM Per Channel
NIC	10x Intel® Ethernet Network Adapter E810-2CQDA2
NIC Firmware Version	3.20 0x8000d846 1.3146.0
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)
Kernel GRUB command	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 + patch
Test date	April 2022
Test by	Intel

3.1 Deployment Setup

For this test we have chosen one of the Networking SKU offerings: pre-production 4th Gen Intel Xeon Scalable Processor. This CPU has 52 cores, 1.7GHz base, and 4GHz Turbo CPU frequency. We also used the Intel® Ethernet Network Adapter E810-2CQDA2 (NICs) that can achieve up to 200Gbps of total bandwidth in systems[3] that are PCIe 4.0 compliant. Each QSFP28 port supports up to 100Gbps, providing the functionality and throughput of two 100Gbps adapters in a single bifurcated PCIe 4.0 x16 slot. An IXIA* hardware traffic generator with 100GbE connection support was used to deliver network packets to the system. On the system under test two Ethernet network adaptors were populated on each CPU socket and several VPP WireGuard gateway instances were executed on each socket. Each VPP instance on CPU socket 0 would have a peer on socket 1 where a WireGuard tunnel is established between the two instances in both directions, i.e., from Gateway A to Gateway B and from Gateway B to Gateway A such that both systems are performing encryption and decryption.

In this paper, we show the VPP WireGuard performance when packet encryption is executed by the CPU cores and when encryption is asynchronously offloaded to Intel QAT. An improvement in the Intel® Xeon® Scalable processor over previous generations is that now Intel QAT comes integrated into the CPU socket! We take advantage of this fact in our test by allocating CPU resources in a completely NUMA-aware fashion–each VPP instance gets CPU cores, Memory, Ethernet network adapters, and Intel QAT devices from the same socket avoiding unnecessary high latency cross-socket data movement. When using Intel QAT devices, we allocate two sibling CPU threads of one CPU core to each Intel QAT device such that one thread is handling encryption, and another thread is handling decryption. When testing the setup without Intel QAT, we keep the core configuration the same for a fair comparison.

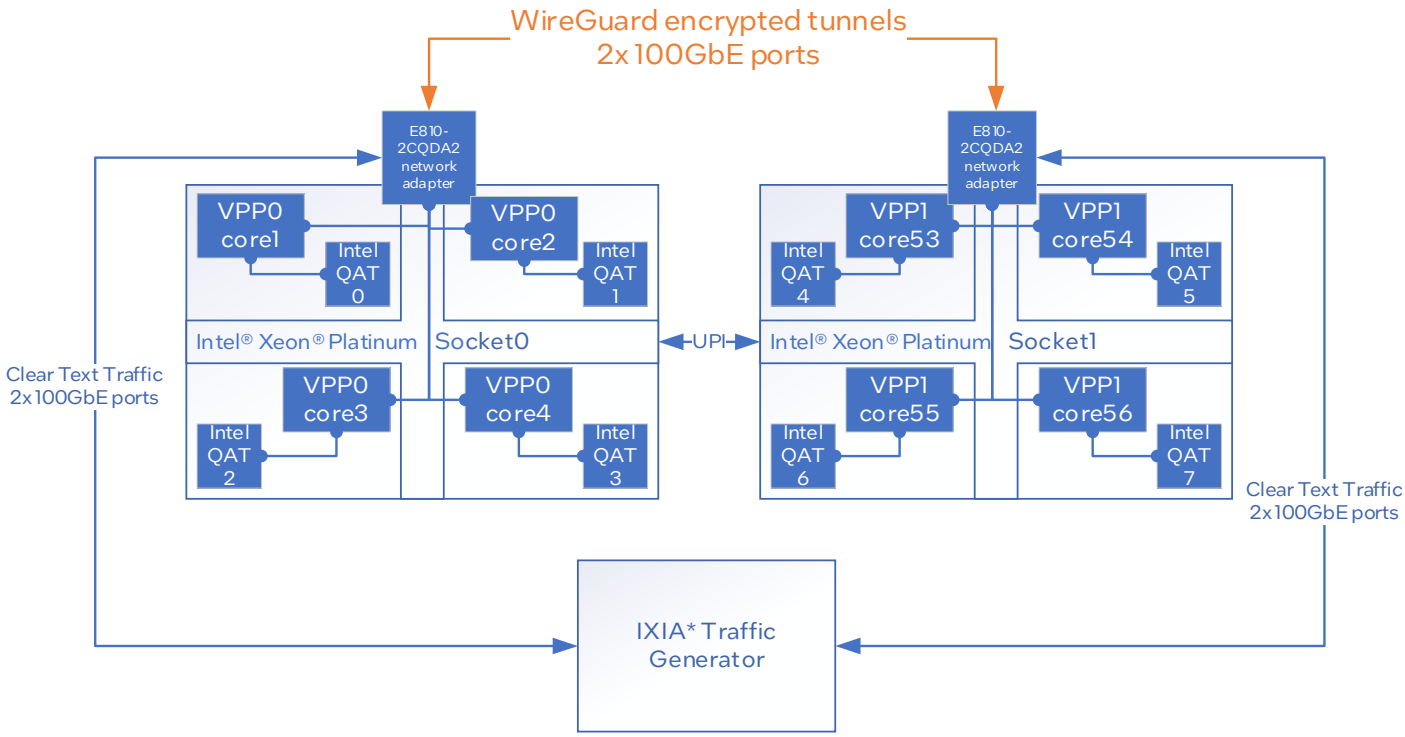

Figure 3. WireGuard performance test setup diagram and CPU resource allocation for the VPP WireGuard gateways

[3] Requires x16 slot bifurcation

3.1.1 Traffic Configuration

Each CPU socket on the system under test was equipped with two Intel® Ethernet Network Adapter E810-2CQDA2 (NIC), and bottom Ethernet ports of each NIC were connected to each other to establish a local network. The top ports of each NIC were connected to the hardware traffic generator. As shown in Figure 2, the Ixia traffic generator sent preconfigured streams of clear packets to each gateway, and the gateways encrypted the packets and forwarded them to their respective peer gateway.

Our test was configured to have two WireGuard tunnels between gateways A and B–one tunnel initiated by each gateway. Ixia traffic generator sends two continuous streams of IPv4 network packets equally spaced in time. The packets are built such that their destination IPv4 headers match the configuration of the underlying networking stacks to route the traffic through the WireGuard tunnels. The configuration of the gateway software is described in the next two sections.

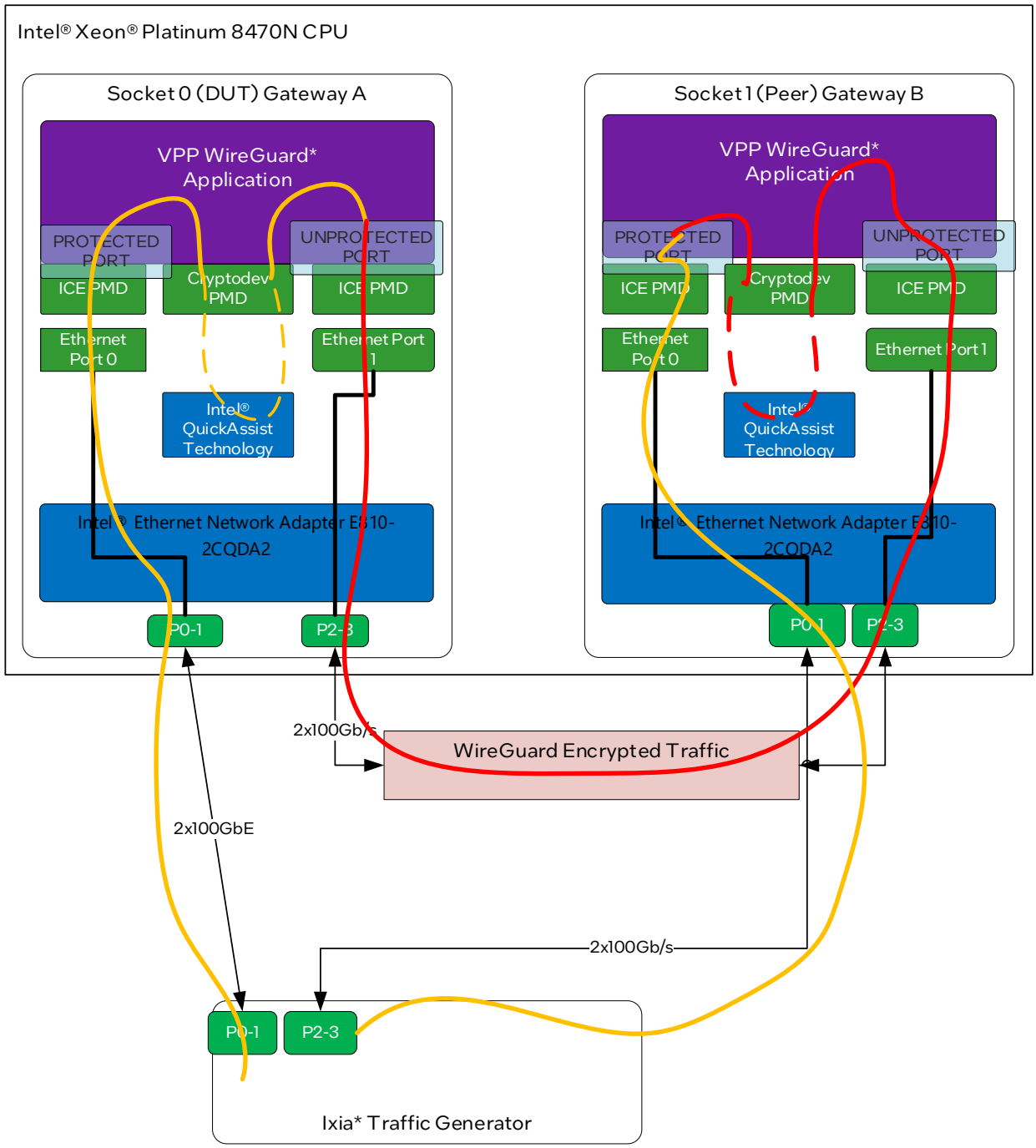

Figure 4. WireGuard test configuration

3.2 VPP Application Configuration

3.2.1 VPP Startup.conf

A VPP process requires a dedicated "startup.conf" file containing specific configurations for the VPP application to run as desired. Since we have two VPP processes running in parallel, two startup.conf files need to be configured, respectively. In Figure 5 we provided a sample startup.conf for VPP0 process. The other startup.conf files content is like Figure 5 but with different "corelist-workers" and DPDK port configuration section.

```
unix {
#Point to the CLI configuration file
        exec /tmp/VPP-WireGuard_SPR_2p1c1t1f_s0_0.00001loss_native-mb-ep0.cfg
        nodaemon
}
cpu {
        main-core 2
# We use one worker core
        corelist-workers 1
}
buffers { buffers-per-numa 5555 }
dpdk {
        no-tx-checksum-offload
        dev default{
                num-tx-desc 512
                num-rx-desc 512
        }
#This is the port connected to Ixia
        dev 0000:91:00.0
        {
                workers 0
        }
#This is the port connected to the second gateway
        dev 0000:94:00.0
        {
                workers 0
        }
        no-multi-seg
}
memory { main-heap-size 1G }
```

Figure 5. VPP Startup.conf

3.2.2 VPP CLI Commands

Startup configuration file in Figure 5 points to a CLI configuration file, which contains VPP software configuration commands required to enable packet forwarding and WireGuard tunnels. As shown in Figure 6, VPP process will receive plain IPv4 packets from one traffic generator port and WireGuard protected UDP packets from one peer port. The packets received from the traffic generator will be forwarded to the virtual WireGuard interface, encrypted, and encapsulated by the WireGuard VPP plugin and routed to the peer system.

```
set interface state HundredGigabitEthernet91/0/0 up
set interface state HundredGigabitEthernet94/0/0 up
set interface mtu packet 2024 HundredGigabitEthernet91/0/0
set interface mtu packet 2024 HundredGigabitEthernet94/0/0
set interface ip address HundredGigabitEthernet91/0/0 64.0.0.1/24
set interface ip address HundredGigabitEthernet94/0/0 255.0.0.128/24
set int promiscuous on HundredGigabitEthernet91/0/0
set int promiscuous on HundredGigabitEthernet94/0/0
# we set static ARP entries, but it's not required
set ip neighbor HundredGigabitEthernet94/0/0 192.168.100.2 00:11:11:11:00:11
set ip neighbor HundredGigabitEthernet91/0/0 64.0.0.2 50:54:00:e1:00:11
set ip neighbor HundredGigabitEthernet94/0/0 255.0.0.129 50:54:00:e1:00:11
# set second Ethernet port as WireGuard tunnel port
ip route add 192.168.100.2/32 via 192.168.100.2 HundredGigabitEthernet94/0/0
wireguard create listen-port 51820 private-key 4BRmziKjgp+BSuwq69z0BI55va0kKHwU5ddjfWwaeGQ= src
192.168.100.1
wireguard peer add wg0 public-key 6BfYHJ77nXWyg6RoL9egv7dK8oK1szE55nft7coqJ08= endpoint
192.168.100.2 allowed-ip 104.0.0.0/24 port 51820 persistent-keepalive 256
set interface state wg0 up
set int ip address wg0 192.168.100.1/24

ip route add 104.0.0.0/24 via 104.0.0.1 wg0
ip route add 192.168.100.1/32 via wg0
ip route add count 1 004.0.0.0/32 via 64.0.0.2 HundredGigabitEthernet91/0/0
# use software encryption with vAES instructions
set crypto handler all ipsecmb
```

Figure 6. VPP CLI Commands for VPP process

3.2.3 VPP CLI Commands to Enable Intel® QAT Crypto Offload

To enable Intel® QAT Crypto to accelerate Chacha20-Poly1305 cryptographic operation, an extra two CLI commands are to be used. They are:

```
# Assign DPDK cryptodev engine to Accelerate Chacha20-Poly1305
set crypto async handler chacha20-poly1305 dpdk_cryptodev
# Enable WireGuard Asynchronous Mode
set wireguard async mode on
```

This configuration would perform the same packet processing operations but offload encryption to Intel® QuickAssist Technology Gen 3 on the CPU.

4 Results

After the system setup is complete, VPP applications are up and running with all commands injected. We started the IXIA* traffic generator to transmit the flows defined in sections 3.1.1 and 3.2.2. We used IXIA built-in RFC2544 test to test our solution. RFC2544 was configured with binary search with 20 second iterations with 1% allowed packet loss. Currently, there is a known issue in VPP Wireguard that prevents us from maintaining high throughput at zero packet loss. The throughput is measured by Ixia hardware and represents the performance on one of WireGuard gateways. In fact, the entire system is processing twice the throughput as each packet passes through each WireGuard gateway once. We collected two sets of performance numbers:

- VPP WireGuard encryption and decryption on "gateway A" DUT for packet sizes of 64/72/128/256/512/1024/1280/1420 bytes and IMIX using Intel-ipsec-mb library to process Chacha20-Poly1305 cryptographic processing.
- VPP WireGuard encryption and decryption on "gateway A" DUT for packet sizes of 64/72/128/256/512/1024/1280/1420 bytes and IMIX using onboard Intel QAT cryptographic accelerator to process Chacha20-Poly1305 cryptographic processing.

Our IMIX distribution was configured as a stream where for each one 1420B packet there are seven 64B packets, and four 594B packets. We chose our largest frame size to be 1420B to avoid the packet from becoming larger than the system's maximum transfer unit (MTU) of 1500B. The current VPP WireGuard implementation does not support frames larger than MTU.

Both solutions were configured to use the same number of CPU cores – four cores and their hyper threads - while the remaining cores were left idle. In the software solution, the CPU cores perform packet processing functionality, such as packet receive and transmit functions, routing table lookup, WireGuard tunnel encapsulation, as well as the ChaCha20-Poly1305 authenticated

encryption. In the hardware solution, the cores perform the same functions but offload the encryption asynchronously to Intel QAT using the DPDK Cryptodev PMD. Asynchronous offload means that after submitting a group of packets for encryption the cores can continue to process subsequent packets while Intel QAT does the heavy lifting.

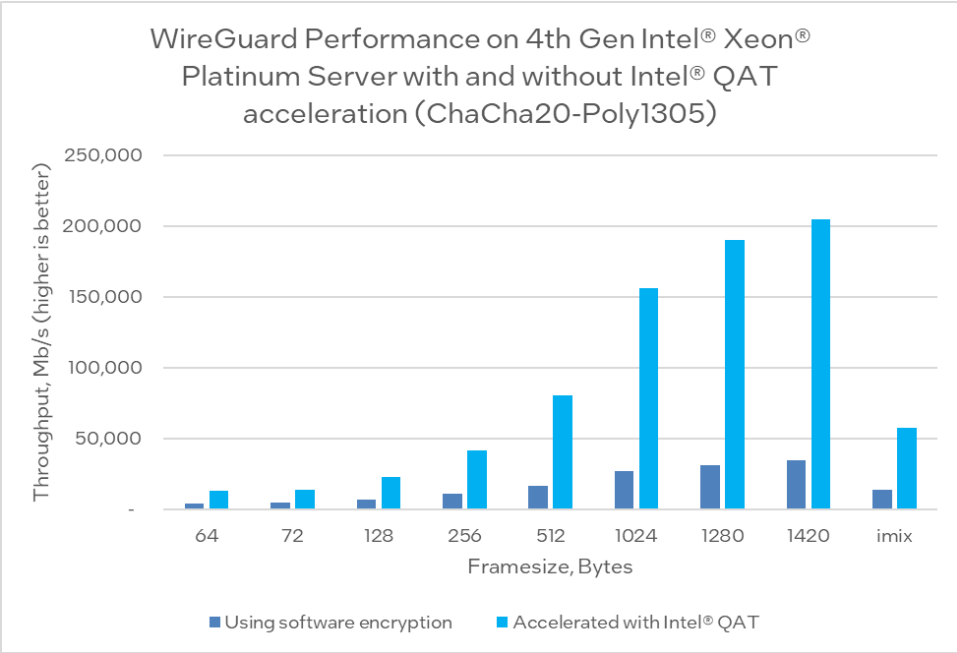

Figure 7. Packet processing performance of VPP WireGuard solution with and without Intel® QAT offload.

As shown in Figure 7, VPP WireGuard solution benefits significantly from offloading the ChaCha20-Poly1305 authenticated encryption to Intel® QAT hardware. We can see that performance benefits of Intel QAT are significant for streams of all packet sizes but are especially pronounced with large packets. VPP WireGuard reached throughput of 204Gb/s with 1420B stream when offloading acceleration to Intel QAT, which is 5.9 times more than the throughput of 34Gb/s when executing all packet processing on the CPU cores. With 256B packets Intel® QAT solution performs up to 3.8 times faster compared to software encryption. Similarly, for IMIX stream (Internet Mix Traffic) we achieved up to 4.2 times more throughput with Intel® QAT acceleration.

5 Summary

This guide demonstrates how to configure and test a VPP based WireGuard Gateway accelerating ChaCha2020-Poly1305 authenticated encryption using Intel AVX-512 instructions (leveraged by Intel-ipsec-mb library) and Intel QAT provided in the latest 4th Generation Intel Xeon Scalable processor.

This guide details the underlying technologies, the challenges encountered, and the hardware and software configurations used to achieve this throughput. 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 improve their WireGuard throughput.

End users are encouraged to test these latest technologies for themselves to evaluate the performance improvements available in the Intel AVX-512 instructions and Intel QuickAssist Technology is provided in the latest 4th Generation Intel Xeon Scalable processor.

Appendix A System BIOS Settings

Socket config Menu	Sub-menu	Sub-menu2	Setting
IIO Configuration	port configuration		x8x8 for all ports
	Intel VT for Directed IO		Enable
Advanced Power Management Configuration	CPU P state Control	Speed Step	Disable
		AVX Licence Pre-Grant	Enable
		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

0123/DN/WIT/PDF 764524-001US