TECHNOLOGY GUIDE

Intel Corporation

Intel® AVX-512 and Intel® QAT - Accelerate WireGuard Processing with Intel® Xeon® D-2700 Processor

Authors

Fan Zhang

Georgii Tkachuk

Pablo De Lara Guarch

Tomasz Kantecki

1 Introduction

WireGuard is a new and fast-growing more secure communication protocol that incorporates increased simplicity, usefulness, and security for the VPN. WireGuard is officially integrated in Linux, Windows, and FreeBSD Kernel and can be implemented in C, Golang, and Rust. Compared to Internet Key Exchange (IKE) / Internet Protocol Security (IPsec), WireGuard has the following advantages:

- Simple:
  - Supports only tunnel mode
  - Supports only single security algorithm for each communication stage
  - No security policy negotiation
- Agile:
  - Fast connect and reconnect two peers
  - No multiple message policy exchange and negotiation as IKEv2
- Modern:
  - State-of-art Curve25519 as PKE algorithm for key exchange
  - Chacha20-Poly1305 for encryption and authentication
  - BLAKE2 for cryptographic hash operation

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

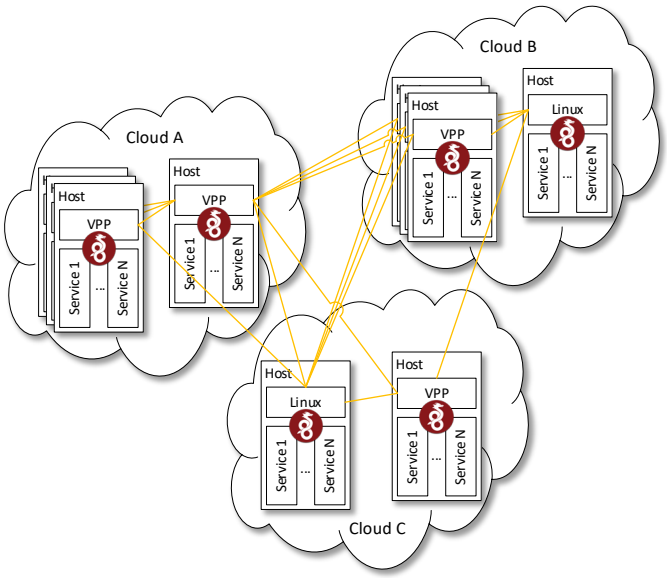

Figure 1. WireGuard Securing Nodes Within and Across Cloud Locations

This guide explains how the latest Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instructions and Intel® QuickAssist Technology (Intel® QAT) Gen 3 enabled in the Intel® Xeon® D-2700 processor can be used to achieve between 3x and 10.5x performance throughput than WireGuard with Linux Kernel WireGuard and routing. Section 3 describes the configuration used to conduct the performance benchmarking.

This document is intended for anyone looking to develop or deploy a more state-of-the-art VPN solution in their existing network infrastructure. The technologies enabled here can be used as a reference point for helping to improve performance in any WireGuard 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 and the Intel® Xeon® D-2700 and D-1700 Processor Experience Kit, which is available at https://networkbuilders.intel.com/intel-technologies/intel-xeon-d-processor-experience-kits.

Table of Contents

Figures

Tables

Document Revision History

REVISION	DATE	DESCRIPTION
001	March 2022	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)
DDP	Dynamic Device Personalization
DSCP	Differentiated Services Code Point
FD.io	Fast Data Input/Output
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
IPsec-MB	Intel® Multi-Buffer Crypto for IPSec
PKE	Public Key Encryption
RSS	Receiver Side Scaling
VAES	Vectorized Advanced Encryption Standard
VAPI	The VPP Binary API
VPP	Vector Packet Processing
WireGuard	Encrypted Virtual Private Network (VPM) Communication Protocol

1.2 Reference Documentation

Table 2. Reference Documents

REFERENCE	SOURCE
AVX-512 Overview	https://www.intel.com/content/www/us/en/architecture-and-technology/avx-512-overview.html
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
Intel® QuickAssist Technology	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
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

REFERENCE	SOURCE
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 Overview

This document is intended as a guide to show how WireGuard throughput may be significantly increased with the hardware and software optimization done to the Intel Xeon D-2700 processor. The hardware optimization is achieved through the Intel AVX-512 instructions utilization and symmetric crypto encryption/decryption done by Intel® QuickAssist Technology (Intel® QAT) Gen 3. 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 utilizes the new Intel AVX-512 instructions for Chacha20-Poly1305 AEAD cryptographic encryption/decryption, powered by the latest Intel® Multi-Buffer Crypto for IPSec (intel-ipsec-mb) library.

In this document, we describe how the test system was set up and deployed using the technologies listed earlier, 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 Gen 3 hardware-accelerated VPP WireGuard implementation.

While this guide goes into detail on how this performance improvement was achieved, the system setup and technology enablement demonstrated in this guide are intended as a reference for anyone trying to improve their networking or WireGuard performance. This guide describes a VPP WireGuard implementation that takes advantage of both Intel AVX-512 instructions and Intel QAT Gen 3 hardware Chacha20-Poly1305 cryptographic operation offloading and showcases the WireGuard performance achieved in the Intel Xeon D-2700 processor.

2.1 Technology Description

2.1.1 Intel Xeon D-2700 Processor

The Intel Xeon D-2700 processor family comes with many new technologies enabled, designed to help improve and increase networking workload performance.

Compared to earlier generation Intel Xeon D processors, the new Intel Xeon D-2700 processor has up to 1.5x Ethernet processing capability, 4x Ethernet connectivity, and 2x PCIe throughput. Most importantly, the new Intel Xeon D-2700 processor achieves significant performance improvement with new Intel® architecture containing new Intel AVX-512 vector instructions to accelerate packet processing as well as cryptographic operations. The new architecture can process up to 100 Gbps Ethernet traffic, with the matching cryptographic processing power brought by built-in Intel QAT Gen 3.

For more information on Intel AVX-512, refer to:

- https://www.intel.com/content/www/us/en/architecture-and-technology/avx-512-overview.html
- https://newsroom.intel.com/articles/crypto-acceleration-enabling-path-future-computing/#gs.ogwf7m
- https://www.intel.com/content/www/us/en/architecture-and-technology/crypto-acceleration-in-xeon-scalable-processors-wp.html
- https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-packet-processing-with-intel-avx-512-instruction-set-solution-brief
- https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-instruction-set-for-packet-processing-technology-guide
- https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-writing-packet-processing-software-with-intel-avx-512-instruction-set-technology-guide

2.1.2 Intel QuickAssist Technology Gen 3

Intel QAT Gen 3 provides hardware acceleration to assist with the performance demands of securing and routing internet traffic and other workloads, such as compression and wireless 3G and 4G LTE algorithm offload, thereby reserving processor cycles for application and control processing. The Intel Xeon D-2700 processor contains the built-in Intel QAT Gen 3 accelerator that assists processing up to 100 Gbps symmetric crypto workload offloading, such as Chacha20-Poly1305 AEAD algorithm processing, so that the CPU can be freed to process the rest of the WireGuard stack processing as well as the remaining application needs.

2.1.3 Intel® Ethernet 800 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 800 Series adapters contain 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 about the Intel Ethernet Controller E810-2CQDA2, refer to:
https://ark.intel.com/content/www/us/en/ark/products/series/184846/intel-ethernet-network-adapter-e810-series.html

2.1.4 Intel Multi-Buffer Crypto for IPSec Library

The intel-ipsec-mb library is a family of highly optimized software implementations of symmetric cryptographic algorithms. With the rich and easy-to-use APIs provided by the intel-ipsec-mb library, you can easily make full use of the CPU latest cryptographic accelerations provided by Intel, including the new Intel AVX-512 vector instructions. The intel-ipsec-mb library provides an optimized implementation of Chacha20-Poly1305, utilizing Intel AVX-512 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 1 KB 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 with IFMA instructions and full reduction is performed only at the end of the message, to get the digest.

For more detailed information about the 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

2.1.5 Fast Data Input/Output, Vector Packet Processing

Fast Data Input/Output (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.

VPP processes the packet in a 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 graphs in VPP. 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.

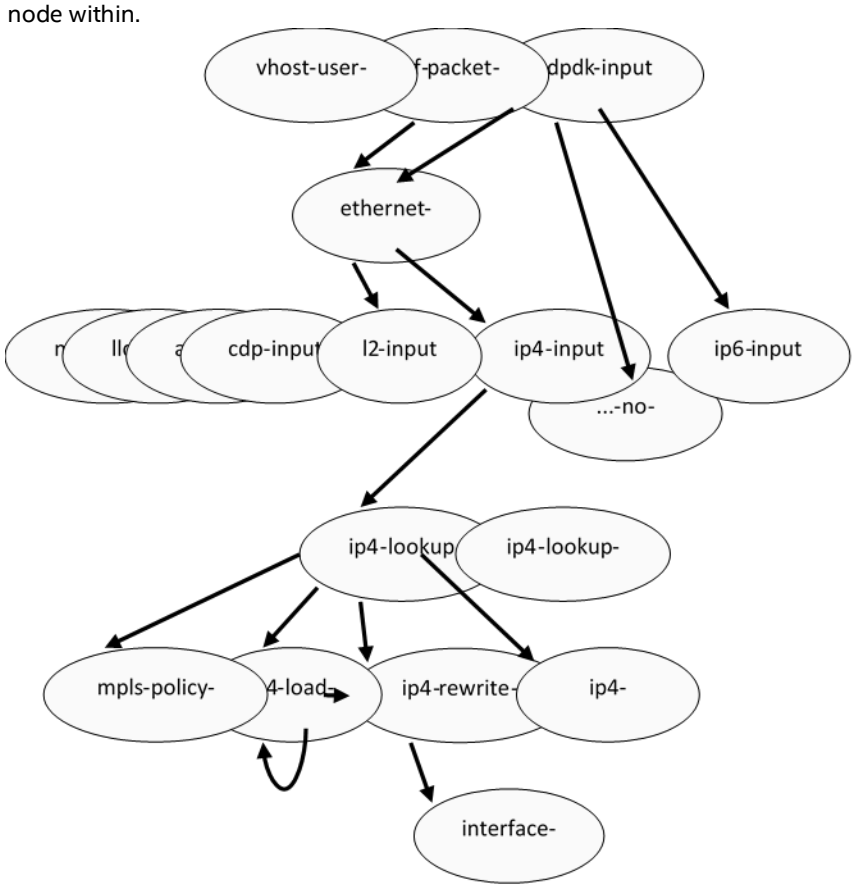

Figure 2. VPP Packet Processing Graph

For more information about VPP, refer to:

- https://wiki.fd.io/view/VPP
- https://s3-docs.fd.io/vpp/22.02/developer/extras/vpp_config.html
- https://s3-docs.fd.io/vpp/22.06/cli-reference/gettingstarted/index.html

2.1.5.1 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 you to configure WireGuard interfaces and peers in addition to existing rich routing and packet RX/TX functionalities. It 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.

2.1.5.2 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. To date, there are three crypto engines:

- Native Engine: The crypto engine specifically designed for VPP that achieves high crypto processing efficiency but with limited algorithms supported. Some Intel AVX-512 accelerations such as vAES and vPCLMUL are automatically enabled if the application is running on the latest architecture CPUs.
- IPsecMB Engine: Integration layer to the 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 the OpenSSL library, with the most comprehensive crypto algorithm support list, but it is least performant.

To maximize the performance of the crypto accelerators, VPP crypto infrastructure also supports an asynchronous working mode such that the workload is enqueued to the crypto accelerator, such as Intel QAT Gen 3, and the next packets are immediately processed, instead of waiting for the current workload to complete processing. Instead, a function block running in a polling manner queries if the previously enqueued workload is processed. This asynchronous working mode helps maximize Intel QAT Gen 3 and CPU packet-processing efficiency. The DPDK Cryptodev engine is one of the supported VPP crypto engines that works in this mode and Intel QAT Gen 3 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 operations of specific algorithms is 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

The following table lists the system setup for the tests described in this guide.

Table 3. System Setup

ITEM	DESCRIPTION
Server Platform	Xeon-D 2700 uATX Customer Reference Board
CPU	Intel Xeon D-2798NX Processor @2.10 GHz
Memory	64 GB: 16 GB x 4 DIMMs @ 2933 MHz DDR4
NIC	Intel® Ethernet Network Adapter E810-2CQDA2 x 1
NIC Firmware Version	V3.0
BIOS	IDVL CRB 1.86B.0020.D15.2107151720
Microcode	0x000136
Operating System	Ubuntu 20.04.3 LTS
Linux Kernel Version	5.4.0-67

ITEM	DESCRIPTION
Kernel GRUB Command	`hugepagesz=1G hugepages=8 default_hugepagesz=1G isolcpus=1-15,17-31 rcu_nocbs=1-15,17-31 nohz_full=1-15,17-31 panic=30 init=/sbin/init net.ifnames=0 image_name=/images/pma-seed-20210319-210608.cpio.gz nmi_watchdog=0 audit=0 nosoftlockup hpet=disable mce=off tsc=reliable numa_balancing=disable memory_corruption_check=0 workqueue.power_efficient=false module_blacklist=ast,iavf modprobe.blacklist=ice init_on_alloc=0 initrd=/kernel/initrd.img-5.4.0-67-generic`
ICE Driver Version	1.6.7
DDP OS Package	1.3.26.0
Kernel WireGuard Version	1.0.20200513-1~20.04.2
VPP Version	21.10-RC0

3.1 Deployment Setup

To test performance of the abovementioned WireGuard solutions, we used two Intel Xeon D-2798NX processors and an Ixia hardware traffic generator with 100 GbE connection support. This test setup is depicted in Figure 3. The first system (Gateway A) serves as the device under test. The second system (Gateway B) serves as the WireGuard peer. The WireGuard connection is established 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. To increase reliability of the test, the two Intel systems were configured to replicate one another as much as possible with respect to hardware specifications and software versions and configurations. We used Intel Xeon D-2798NX CPUs for both gateways and populated the same number of memory DIMMs of the same kind into each system.

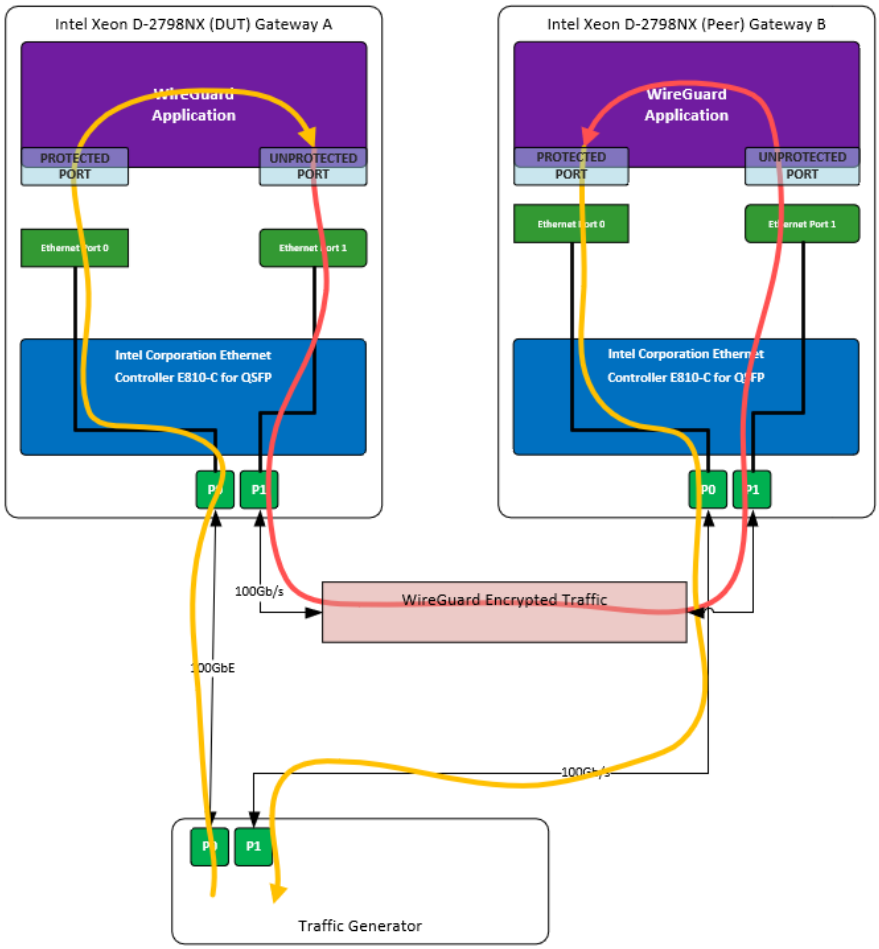

Figure 3. WireGuard Performance Test Setup Diagram

3.1.1 DUT NIC Ports and Ixia Ports Connection

Each system was equipped with one Intel Ethernet Network Adapter E810-2CQDA2 (NIC). The 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. All Ethernet ports used in this test were 100 Gigabit Ethernet (GbE) ports. As shown in Figure 3, 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.

3.1.2 Ixia Flow Configuration

Our test was configured to have two WireGuard tunnels between Gateway A and Gateway B – one tunnel initiated by each gateway. The 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. Figure 4 lists the Ixia configuration details of the traffic stream sent to one of the DUT ports. These traffic streams consist of IPv4 packets with destination MAC addresses matching the MAC address of the neighboring Ethernet port and destination IP address matching the IP of the destination Ixia port. Configuration of the gateway software is described in the next two sections.

Name	Value
Frame	length: 64
Ethernet II	
Ethernet Header	
Destination MAC Address	50:54:00:e0:01:00 [Inc: 50:54:00:e0:01:00, 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	4.0.0.0
Destination Address	8.0.0.0
IP options	
Payload	Increment Byte
Ethernet II (Trailer)	
Frame Check Sequence CRC-32	0x<Auto>0

Figure 4. Ixia Traffic Configuration

3.2 Linux Kernel WireGuard Configuration

The OS network software was configured to establish appropriate routes according to the diagram in Figure 5. We installed WireGuard on our Ubuntu Linux with apt package manager and configured it according to the WireGuard Quick Start guide.

```
apt install wireguard
ip link add dev wg0 type wireguard
ip address add dev wg0 192.168.2.1/24
wg genkey > private
wg pubkey < private > publickey
wg set wg0 listen-port 51820 private-key ./private peer
Nfcx065/ehwTi+F1uXCVPVRpOY5jFkjUCGffSDAhGiQ= allowed-ips
8.0.0.0/24,192.168.2.0/24,4.0.0.0/24 endpoint 172.16.0.2:51820
ip link set wg0 up



apt install wireguard
ip link add dev wg0 type wireguard
ip address add dev wg0 192.168.2.2/24
wg genkey > private
wg pubkey < private > publickey
wg set wg0 listen-port 51820 private-key ./private peer
h8Es+0U72W9d7ViUs0c9GhgCo09eTLZj81baqDyhexM= allowed-ips
8.0.0.0/24,192.168.2.0/24,4.0.0.0/24  endpoint 172.16.0.1:51820
ip link set wg0 up
```

Figure 5. Linux Kernel WireGuard Configuration

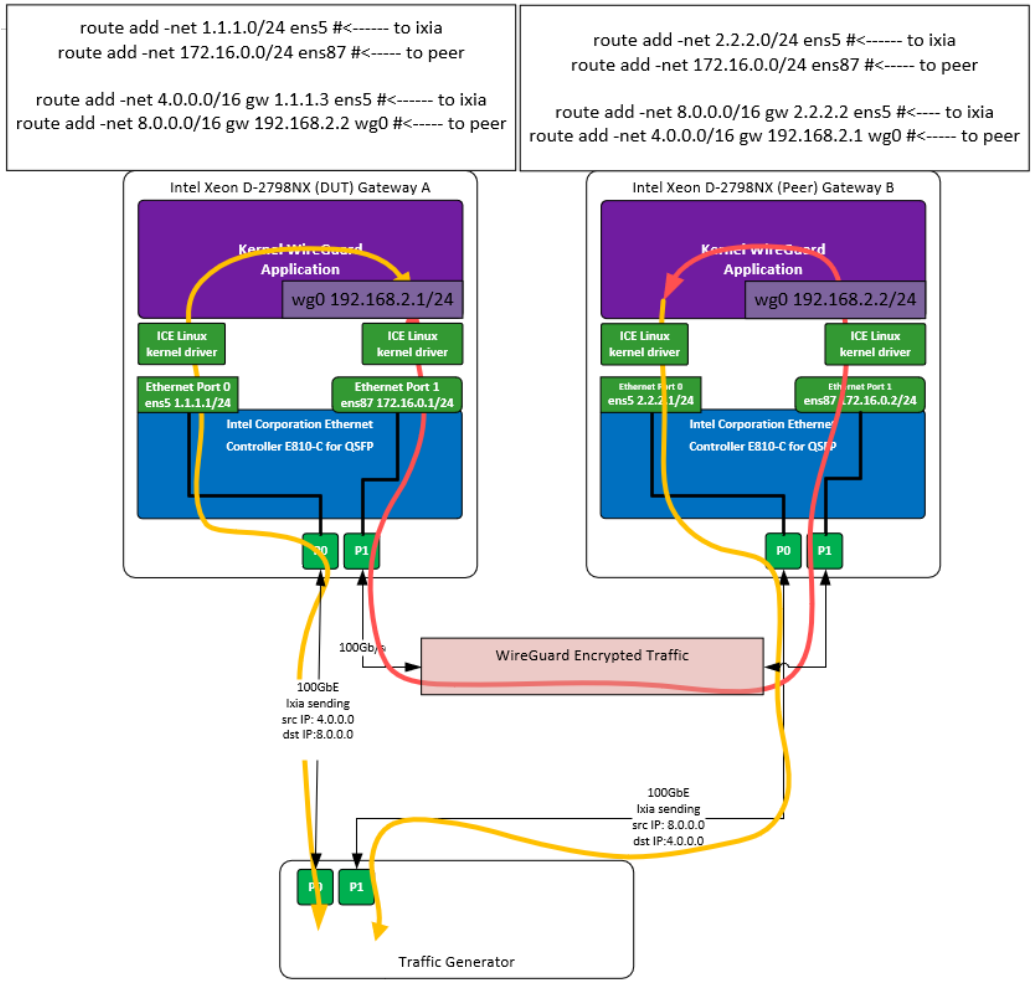

Figure 6. Kernel WireGuard Test Configuration

3.3 VPP Application Configuration

3.3.1 VPP `startup.conf` File

For a VPP application to run as desired, a VPP process requires a dedicated configuration file containing specific configurations being loaded along with the application execution. Since we have two VPP processes running in parallel, two configuration files need to be provided. Figure 7 provides a sample configuration file for the VPP0 process. The other configuration file is similar to Figure 7, but with different values for `corelist-workers` and the DPDK port configuration section. The method to load `startup.conf` files and to start the application can be found in the VPP Configuration File guide.

```
unix {
#Point to the CLI configuration file
        exec /tmp/VPP-WireGuard_ICXD_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 7. VPP0 `startup.conf` File

3.3.2 VPP CLI Commands

The startup configuration file in Figure 7 points to a CLI configuration file that contains VPP software configuration commands required to enable packet forwarding and WireGuard tunnels. As shown in Figure 8, the VPP process receives plain IPv4 packets from one traffic generator port and WireGuard protected UDP packets from one peer port. The packets received from the traffic generator are 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 8. VPP CLI Commands for VPP Process

3.3.3 VPP CLI Commands to Enable Intel QAT Gen 3 Crypto Offload

To enable Intel QAT Gen 3 crypto to accelerate Chacha20-Poly1305 cryptographic operation, add two CLI commands. 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 performs the same packet processing operations but offloads encryption to Intel QAT Gen 3 on the CPU.

4 Results

After the system setup was complete and VPP applications were up and running with all commands injected, we started the Ixia traffic generator to transmit the flows defined in Section 2. The test was set to run for 20 seconds. We collected three sets of performance numbers:

- Linux Kernel WireGuard encryption and decryption on Gateway A DUT for packet sizes of 66/72/128/256/768/1024/1280/1420 bytes
- VPP WireGuard encryption and decryption on Gateway A DUT for packet sizes of 66/72/128/256/768/1024/1280/1420 bytes, both using the intel-ipsec-mb library to process Chacha20-Poly1305 cryptographic processing
- VPP WireGuard encryption and decryption on Gateway A DUT for packet sizes of 66/72/128/256/768/1024/1280/1420 bytes, both using the Intel QAT Gen 3 cryptographic accelerator to process Chacha20-Poly1305 cryptographic processing

When running Linux Kernel WireGuard tests, we observed that packet processing operations such as interrupts, route lookups, and WireGuard operations were distributed across many cores. We did not take additional steps to limit the execution to a single core like we did in VPP because there is no trivial way to configure the Linux WireGuard application to use fewer cores. This resulted in the Kernel WireGuard solution consuming roughly 7 logical processors out of 40 threads available on the system. This CPU utilization number was calculated by adding all CPU utilization values on all threads measured by the Linux `htop` command while the workload was running and 1420 B streams were being delivered to the system. Conversely, our next test showed that VPP used only one physical CPU core at 100% utilization.

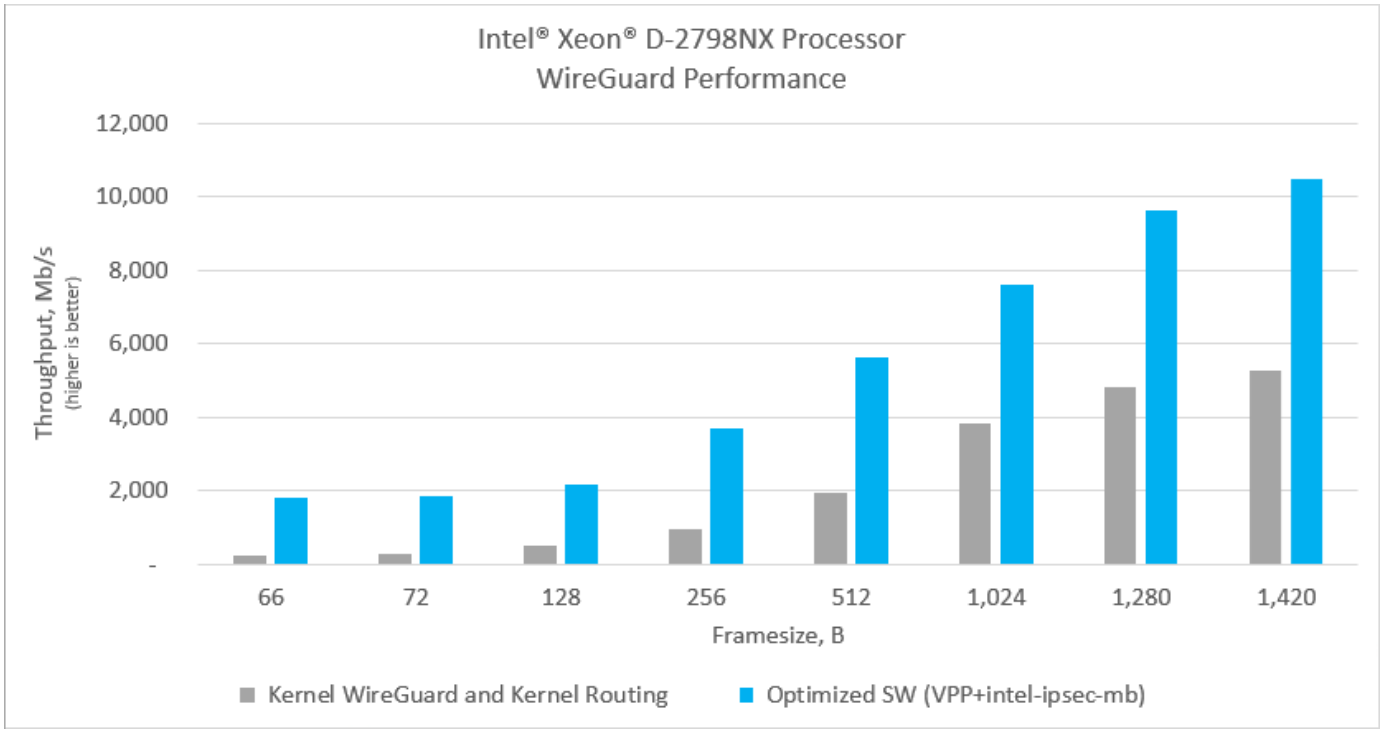

Figure 9. Performance Comparison of Kernel WireGuard Implementation and VPP WireGuard with Software Encryption

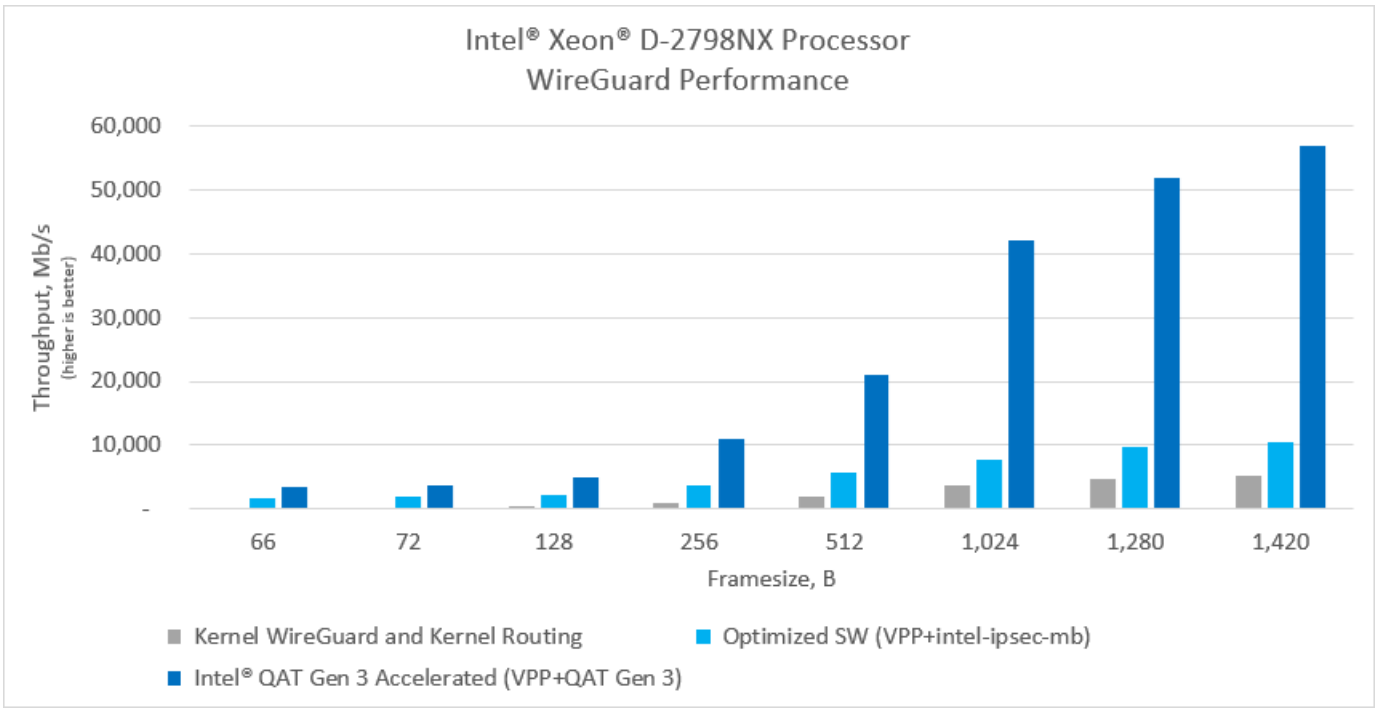

Figure 10. Performance Comparison of Kernel WireGuard, VPP WireGuard with Software Encryption, and VPP WireGuard with Hardware Lookaside Encryption

From Figure 10, compared to the Linux Kernel WireGuard stack and routing running on Intel Xeon D-2798NX processors, the WireGuard single peer encryption and decryption by VPP and intel-ipsec-mb library running on the same platform has 7.2 times more throughput for 66-byte packets, reaching 1,799 Megabits per second (Mb/s), where the Linux Kernel only achieved 250 Mb/s. If VPP offloads the Chacha20-Poly1305 cryptographic processing to Intel QAT Gen 3, the throughput can be further improved to 3,377 Mb/s, which is 13.5 times more throughput than the Linux Kernel. Such advantages of VPP WireGuard with both intel-ipsec-mb CPU optimization and Intel QAT Gen 3 acceleration remain across all packet sizes to 1,420 bytes. In the 1,420 bytes packet size tests, the Linux Kernel WireGuard stack and routing achieved 5,274 Mb/s, the VPP with intel-ipsec-mb optimization achieved

10,486 Mb/s, and the VPP with Intel QAT Gen 3 Chacha20-Poly1305 cryptographic operation offload achieved 56,921 Mb/s, 10.79 times more throughput than the Linux Kernel.

5 Summary

This guide demonstrates how Intel AVX-512 instructions (leveraged by the intel-ipsec-mb library) and Intel QuickAssist Technology Gen 3 provided in the latest Intel Xeon D-2798NX processor can be used to achieve 3x-10x performance increase of WireGuard stack processing and routing performance versus Linux Kernel, reaching 10,486 Mbps and 56,921 Mbps throughput respectively.

This guide has detailed the underlying technologies, the challenges faced, and the hardware and software configurations used to achieve this throughput. And 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.

You are encouraged to test these latest technologies for yourself to evaluate the performance improvements available in the Intel AVX-512 instructions and Intel QuickAssist Technology Gen 3 provided in the latest Intel Xeon D-2798NX processor.

Appendix A System BIOS Settings

Menu (Advanced)	Path to BIOS Setting	BIOS Setting	Value
POWER CONFIGURATION	SOCKET CONFIGURATION -> ADVANCED POWER MANAGEMENT CONFIGURATION -> CPU P STATE CONTROL	EIST PSD FUNCTION	HW_ALL
		AVX License Pre-Grant	ENABLE
		AVX ICCP Pre-Grant Level	512 Light
		BOOT PERFORMANCE MODE	MAX PERFORMANCE
		ENERGY EFFICIENT TURBO MODE	DISABLED
		TURBO MODE	DISABLED
		SPEEDSTEP (P STATES)	DISABLED
	SOCKET CONFIGURATION -> ADVANCED POWER MANAGEMENT CONFIGURATION -> HARDWARE PM STATE CONTROL	HARDWARE P-STATES	DISABLED
	SOCKET CONFIGURATION -> ADVANCED POWER MANAGEMENT CONFIGURATION -> CPU C STATE CONTROL	CPU C6 REPORT	DISABLED
		ENHANCED HALT STATE (C1E)	DISABLED
		PACKAGE C STATE	C0/C1 STATE
	SOCKET CONFIGURATION -> ADVANCED POWER MANAGEMENT CONFIGURATION -> CPU - ADVANCED PM TUNING -> ENERGY PERF- BIAS	UNCORE FREQUENCY SCALING	DISABLED
		UNCORE FREQUENCY RAPL	DISABLED
		POWER PERFORMANCE TUNING	BIOS CONTROLS EPB
		ENERGY_PERF_BIAS_CFG MODE	PERFORMANCE
IIO CONFIGURATION	SOCKET CONFIGURATION -> IIO CONFIGURATION -> IOAT CONFIGURATION	RELAX ORDERING	NO
	SOCKET CONFIGURATION -> IIO CONFIGURATION -> INTEL VT FOR DIRECTED I/O (VT-D)	INTEL VT FOR DIRECTED I/O (VT-D)	DISABLED
	SOCKET CONFIGURATION -> MEMORY CONFIGURATION	ENFORCE POR	POR

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.

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.

0322/DN/WIPRO/PDF 723086-001US