

# Maximizing vCMTS Data Plane Performance with 3rd Gen Intel® Xeon® Scalable Processor Architecture

## Intel platform technologies boost virtualized cable modem termination system (vCMTS) data plane performance by 70%

### Authors

**Brendan Ryan**  
System Architect (Intel)

**Michael O'Hanlon**  
Senior Principal Engineer (Intel)

**David Coyle**  
Senior Software Engineer (Intel)

**Rory Sexton**  
Network Software Engineer (Intel)

**Subhiksha Ravisundar**  
Network Software Engineer (Intel)

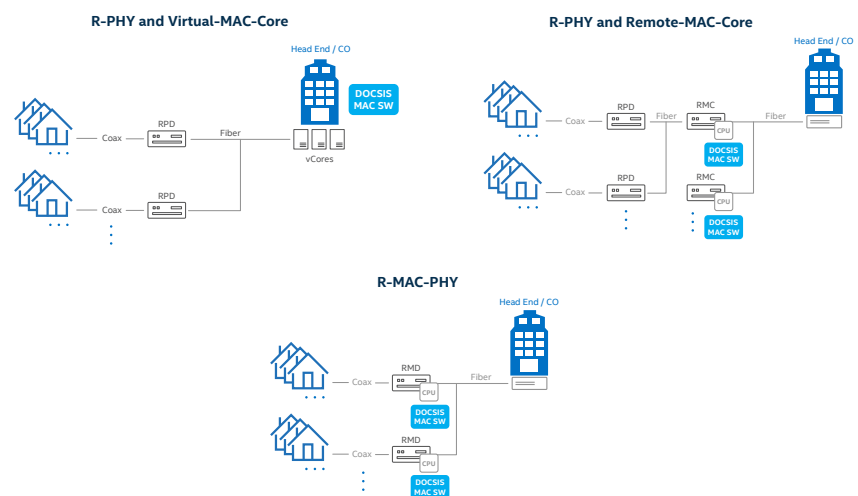
### Introduction

The standardization of distributed access architecture (DAA) for DOCSIS and further advancements in the flexible MAC architecture (FMA) standard have enabled the transition to a software-centric cable network infrastructure. This paper will explore how Intel technologies can be utilized to increase performance for vCMTS for the various deployment scenarios shown below (see Figure 1). In each scenario, the same DOCSIS MAC software may be deployed, whether as:

- a virtual MAC core (vCore), also known as a virtualized cable modem termination system (vCMTS) on a server in a multiple-system operator (MSO) headend,
- part of a remote-MAC-core (RMC) and remote-PHY deployment on an edge compute node,
- or part of a remote-MAC-PHY device (RMD) deployment on an edge compute node

### Table of Contents

Intel vCMTS Reference Data Plane .....	2
vCMTS Data Plane Performance Analysis – Single Service Group ..	3
Crypto and CRC Processing in the vCMTS Data Plane Pipeline.....	4
Using Hyper-threading in the vCMTS Data Plane Pipeline.....	6
System Scalability and Server Sizing .....	8
Compelling vCMTS Performance on 3rd Gen Intel® Xeon® Scalable Processors.....	8
Appendix A .....	10
Appendix B .....	12
Appendix C .....	13
Appendix D .....	13
Appendix E .....	15



**Figure 1. Flexible MAC Architecture (FMA) Deployment Scenarios**

Performance for such a software-centric approach is greatly boosted by technologies such as Data Plane Development Kit (DPDK) and the Intel Multi-Buffer Crypto library (intel-ipsec-mb), which provide highly optimized packet processing in software that is tightly coupled to the ever-developing Intel® architecture. Intel architecture provides native instructions and features that specifically accelerate data plane packet processing for access network functions such as vCMTS.

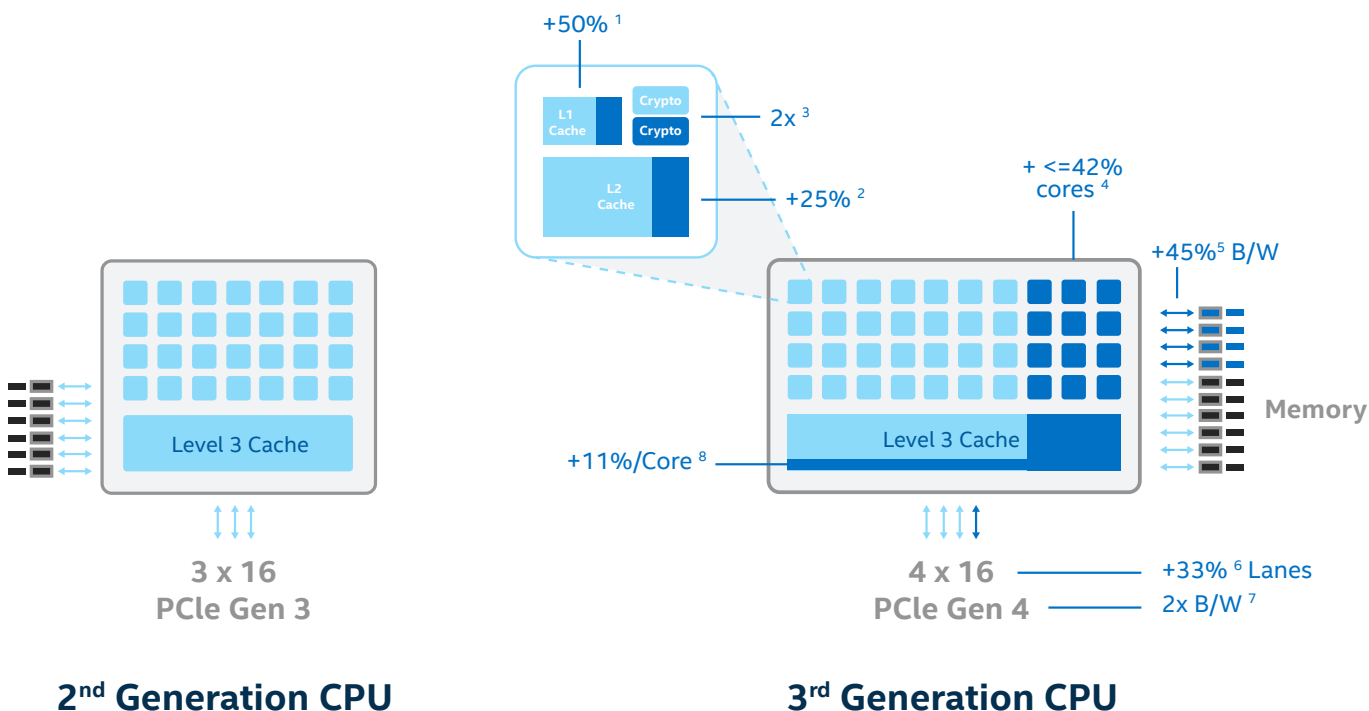


Figure 2. Intel® Xeon® CPU Generational Enhancements

Independent software vendors (ISVs) can significantly improve vCMTS performance on 3rd Gen Intel® Xeon® Scalable processor architecture (codename “Ice Lake”) by taking advantage of the gen-on-gen CPU architecture enhancements shown in Figure 2 which include bigger cache-sizes at each level, higher core-count, more memory channels at higher speed and greater I/O bandwidth.

This paper focusses specifically on how to take advantage of advanced features such as:

- Enhanced Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
- Dual AES encryption engines
- Intel® Vector AES New Instructions (AES-NI)
- Intel® Vector PCLMULQDQ carry-less multiplication instruction
- Intel® QuickAssist Technology

The paper also provides insights into implementation options and establishes an empirical performance data baseline that can be used to estimate the capability of a vCMTS platform running on industry-standard, high-volume servers based on 3rd Gen Intel Xeon Scalable processor architecture. Actual measurements were taken on a 3rd Gen Intel Xeon Scalable processor-based system (see Appendix D for configuration details).

The data demonstrates how a single 3rd Gen Intel Xeon Scalable processor core running at 2.2 GHz clock speed can support the downstream channel bandwidth for close to five orthogonal, frequency-division multiplexing (OFDM) channels in a pure DOCSIS 3.1 configuration for an Internet

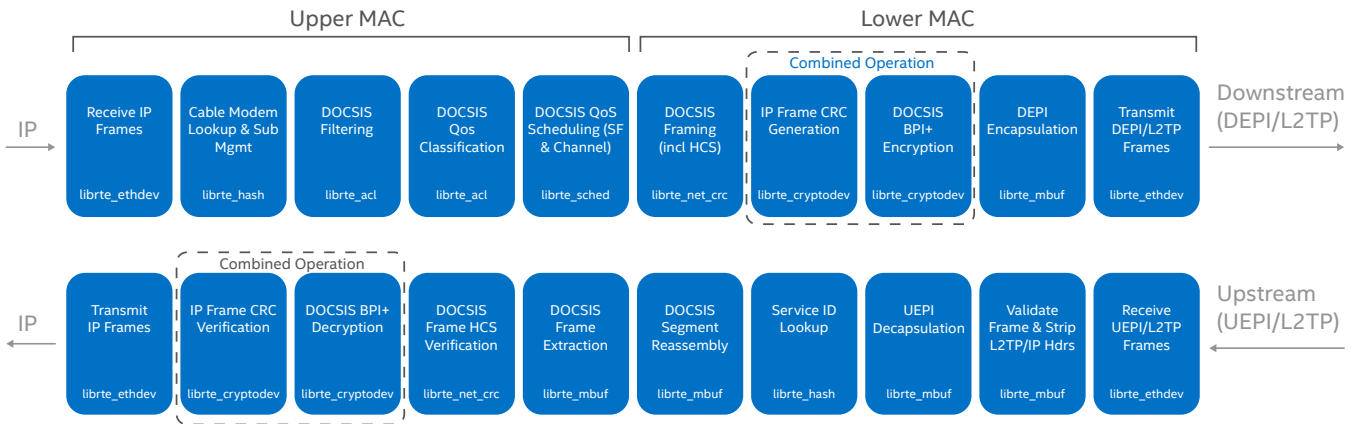
mix (IMIX) traffic blend. When Intel® QuickAssist Technology (Intel® QAT) acceleration is employed in the system, a single core can satisfy close to the maximum downstream bandwidth of a pure DOCSIS 3.1 configuration. Since vCMTS workloads exhibit good scalability, a typical server blade based on dual 3rd Gen Intel Xeon Scalable processors (e.g. with 64 processor cores total) can achieve compelling performance density.

It is worth noting that these performance benefits can also be delivered by deploying a general-purpose compute component in the RMC and RMD scenarios shown in Figure 1. In other words, the same performance benefits of Intel architecture and complementary technologies, such as Intel QAT and the Intel Multi-Buffer Crypto library, can be used to achieve similar performance on an edge compute node.

DOCSIS MAC functionality can be broken down into four categories: downstream data plane, upstream data plane, control plane, and system management. From a network performance perspective, the most compute-intensive workload is data plane processing, and consequently, is the focus of this paper.

#### Intel vCMTS Reference Data Plane

Intel has developed a vCMTS reference data plane that is compliant with DOCSIS 3.1 specifications and based on the DPDK packet processing framework. It is publicly available on the Intel 01.org open-source website at [01.org/access-network-dataplanes](https://01.org/access-network-dataplanes). The main objective of this development is to provide a tool to demonstrate the vCMTS data plane packet processing performance of Intel® Xeon® processor-based platforms and to assist ISVs and MSOs in deploying a vCMTS.



**Figure 3. Intel vCMTS Reference Data Plane**

Figure 3 shows the upstream and downstream packet processing pipelines implemented for the Intel vCMTS Reference Data Plane. The downstream data plane is implemented as a two-stage pipeline, which perform upper MAC and lower MAC processing respectively. The DPDK API used for each significant DOCSIS MAC data plane packet-processing stage is also shown.

A detailed description of the upstream and downstream packet processing stages shown in Figure 3 is provided in Appendix A. Many key innovations and performance optimizations are supported by the Intel vCMTS Reference Data Plane, including the following:

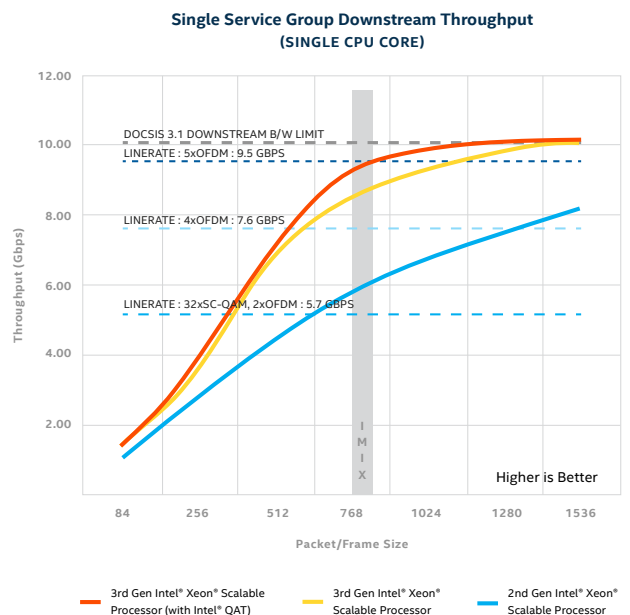
- Optimized multi-buffer implementation of combined AES cryptographic (crypto) and CRC processing based on Intel AES-NI and AVX-512 instructions
- Optimized multi-buffer implementation of DES crypto processing based on Intel AVX-512 instructions
- Acceleration of AES and DES DOCSIS crypto processing using Intel QuickAssist Technology
- Optimized CRC32 and DOCSIS header check sequence (HCS) calculation based on Intel AVX-512 instructions
- DOCSIS service flow and channel scheduling implementation based on DPDK HQoS
- Packet Streaming Protocol (PSP) fragmentation/reassembly implementation based on the DPDK Mbuf API
- DOCSIS MAC data plane pre-processing using the 100G Intel® Ethernet 800 Series network interface card (NIC)
- Configurable DOCSIS MAC data plane threading options – dual-thread, single-thread, separate or combined upstream and downstream threads

The Intel vCMTS Reference Data Plane runs within an industry-standard, Docker container-based environment with Kubernetes orchestration, which is described in detail in Appendix B.

**vCMTS Data Plane Performance Analysis – Single Service Group**

Performance tests were run using the Intel vCMTS Reference Data plane with a channel configuration to demonstrate the maximum throughput capability for a pure DOCSIS 3.1 deployment. Performance with crypto acceleration using both the Intel Multi-Buffer Crypto library and Intel QAT is shown. The performance of the 3rd Gen Intel Xeon Scalable processor running at 2.2 GHz core frequency is also compared to a 2nd Generation Intel® Xeon® Scalable processor (codename “Cascade Lake”) running at 2.3 GHz.<sup>9</sup>

The channel configuration is a pure DOCSIS 3.1 deployment with six OFDM channels that produce a theoretical cumulative bandwidth of 11.3 Gbps; however, the effective downstream bandwidth is limited to 10 Gbps for the DOCSIS 3.1 bandwidth limit per service group (SG). Note that this is enforced by the vCMTS data-plane pipeline with Downstream QoS rate-limiting of 10 Gbps per service group.



**Figure 4. Single-Core Downstream Throughput for Various Packet Sizes**

Downstream throughput is shown in Figure 4 for a range of packet sizes. With increasing packet size, the header-to-payload processing ratio declines, generally resulting in higher overall packet throughput. However, on the other hand, software-based encryption and CRC generation requires more computing resources for larger packet sizes. The IMIX bar shows the measured throughput range when running the IMIX packet mixture. Though the weighted average packet size is approximately 1,000 bytes, empirical measurements suggest that the IMIX packet size performance is in the range of 800 to 900 bytes.

The software-only encryption curve (i.e. no QAT crypto acceleration) for the 3rd Gen Intel Xeon Scalable processor indicates downstream throughput of 9 Gbps at IMIX and DOCSIS 3.1 B/W limit saturation occurring somewhere close to 1,024 byte packet sizes. The QAT accelerated curve is steeper than the software-only curve, reaching 9.6 Gbps at IMIX, representing a performance benefit of Intel QAT crypto acceleration for large packet processing. Note that QAT Crypto acceleration is more beneficial for larger packets as there is a fixed CPU cycle cost of QAT offload which is independent of packet size. Taking into account that CRC processing is done in software, this benefit starts between 256 and 512 byte packet sizes. There are other benefits of this fixed cost such as less jitter related to packet-size variance.

Note also, the software-only curve for the 2nd Gen Intel Xeon Scalable processor indicates downstream throughput of 6.6Gbps at IMIX and DOCSIS 3.1 B/W limit saturation is not achieved for packet sizes smaller than 1,536 bytes. This represents a 35 percent improvement in single-core throughput between these two generations of Intel Xeon processors.

### Crypto and CRC Processing in the vCMTS Data Plane Pipeline

Encryption, specifically AES-based baseline privacy interface (BPI+) encryption, and data packet CRC generation consume a significant portion of CPU cycles in the vCMTS data plane packet-processing pipeline. However, by using the Intel Multi-Buffer Crypto library and Intel QuickAssist Technology acceleration, the performance of this processing has been significantly improved in 3rd Gen Intel Xeon Scalable processors.

Figure 5 illustrates key enhancements to AES Encryption in 3rd Gen Intel Xeon Scalable processor architecture. Crypto performance is significantly improved due to AES crypto enhancements which add a second AES port for each CPU core and support for AVX-512 vectorized AES-NI instructions. Furthermore, a new AVX-512 vector instruction for the PCLMULQDQ carry-less multiplication instruction significantly improves CRC performance, which is another significant part of the DOCSIS MAC data plane processing pipeline.

In a recent innovation Intel has also implemented support for combined crypto and CRC processing in the Intel Multi-Buffer Crypto library and DPDK to further reduce CPU cycle cost and enable significant improvements in DOCSIS MAC data plane performance.

In the case of acceleration provided by Intel QAT, it may be either integrated in a chipset (for some 3rd Gen Intel Xeon processor SKUs) or added to a system via a PCI add-in card. This technology performs hardware-accelerated crypto functions and effectively reduces the CPU cycle cost of encryption regardless of packet size or encryption type (for AES or DES).

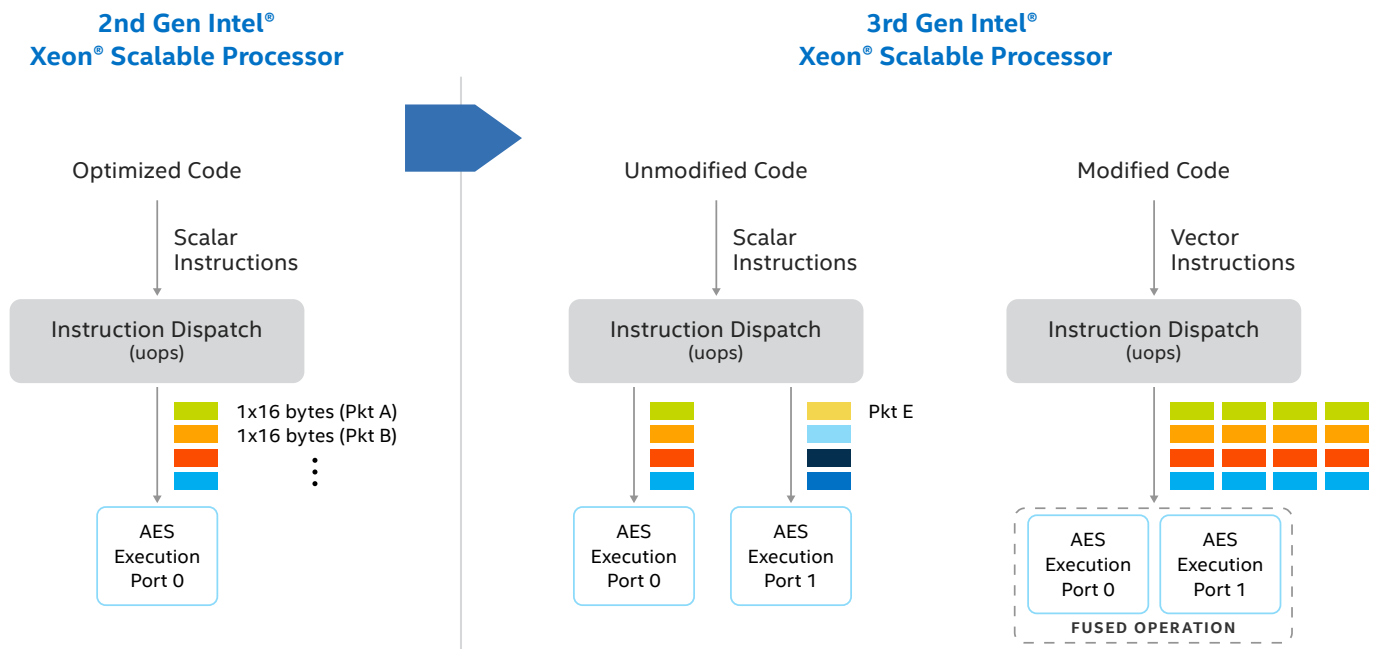
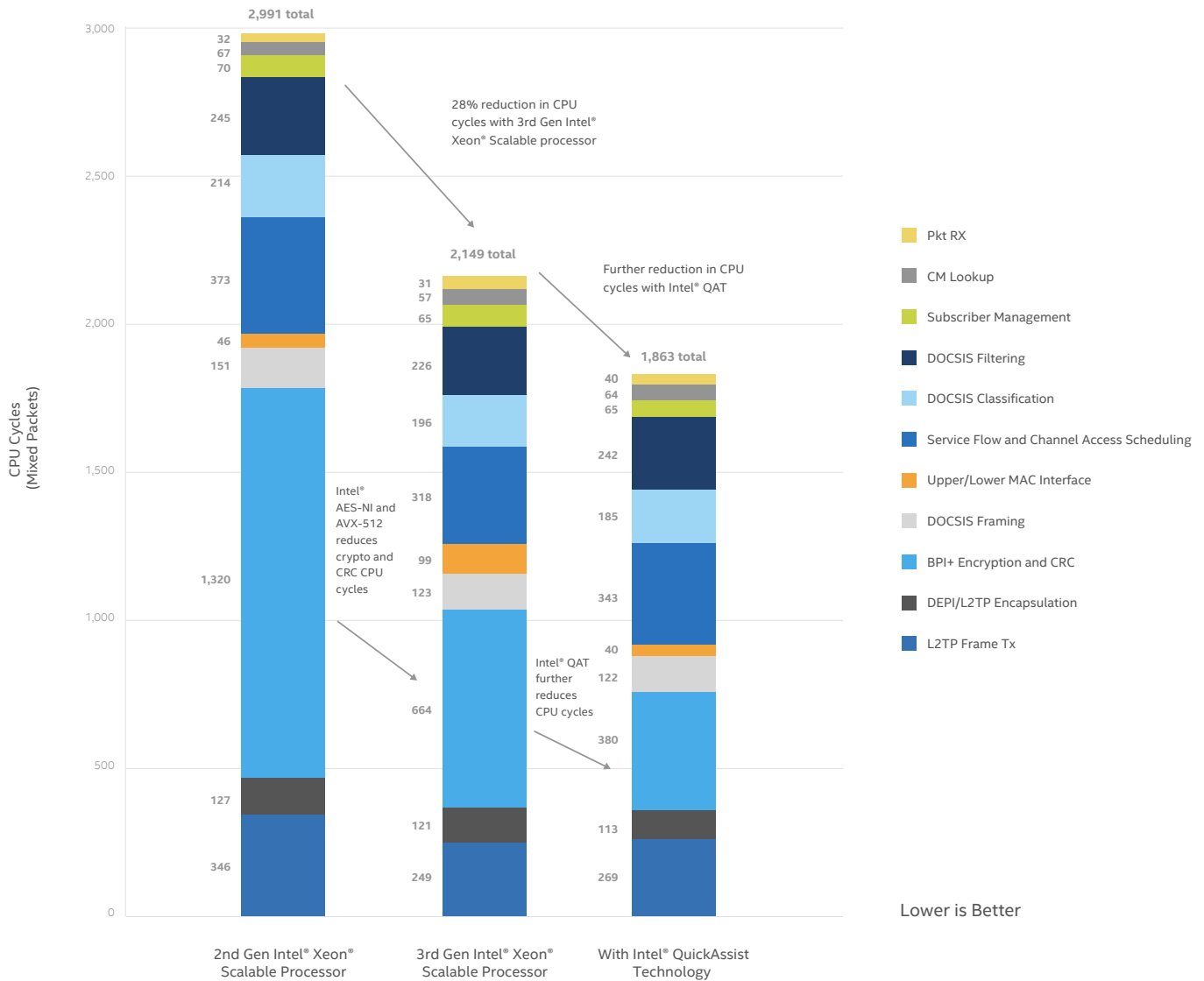


Figure 5. AES enhancements for 3rd Gen Intel® Xeon® Scalable processor architecture

## Downstream



**Figure 6. Downstream Data Plane CPU Cycle Comparison (IMIX)**

Note that as Intel QAT does not yet support combined crypto and CRC processing, the CRC part is still generated separately in software which nonetheless is accelerated by the new vectorized PCLMULQDQ instruction.

In the case of software-based crypto and CRC processing, a key consideration is that it can be characterized as a per-byte cost; hence, the cost of encryption/decryption and CRC generation for a packet is directly proportional to the size of the packet.

As with QAT acceleration, Intel Multi-Buffer Crypto library acceleration is embedded in DPDK and implements data buffer processing techniques which make optimal use of Intel CPU instructions. Starting with version 20.08 of DPDK, the Intel Multi-Buffer Crypto library has been updated with the aforementioned implementation of combined crypto and CRC

processing for DOCSIS. Improved performance is achieved by interleaving Intel AES-NI and PCLMULQDQ instructions in the code, taking advantage of the benefits of processing multiple independent data buffers in parallel.

Figure 6 illustrates the performance improvement of multi-buffer Crypto-CRC in software on 3rd Gen Intel Xeon Scalable processor architecture versus 2nd Gen Intel Xeon Scalable processor architecture, and the further improvement that can be achieved by using QAT acceleration. Performance comparison is shown for the downstream DOCSIS MAC data plane pipeline when processing a downstream-specific IMIX packet size distribution.<sup>9</sup>

Figure 7 shows the Upstream performance improvement due to multi-buffer decryption in software on 3rd Gen Intel Xeon

Scalable processor architecture versus 2nd Gen Intel Xeon Scalable processor architecture. Performance comparison is shown when processing an upstream-specific IMIX packet size distribution. It is assumed that CRC verification is not required on upstream. And note also that for upstream traffic, there is generally a high percentage of small packets so there isn't a significant benefit in using Intel QAT.<sup>9</sup>

Using Hyper-Threading in the vCMTS Data Plane Pipeline

Intel® Hyper-Threading Technology may be used to further improve vCMTS performance. Figure 8 shows the split of functionality between downstream upper and lower MAC data plane processing.<sup>9</sup>

As the Upper MAC and the associated Lower MAC have a similar CPU cycle cost, there is a benefit in deploying them separately on sibling hyper-threads (i.e., hyper-threads of

the same processor core) due to the effective greater number of instructions per second that can be achieved with hyper-thread time-slicing.

The upstream data plane has lower bandwidth requirements than the downstream; however it also benefits from hyper-threading by running two, single-threaded, upstream data plane instances on sibling hyper-threads.<sup>10</sup>

Performance benchmarking of the Intel vCMTS Reference Data Plane has shown that total vCMTS throughput is significantly improved for a dual-thread downstream data plane versus single-thread.

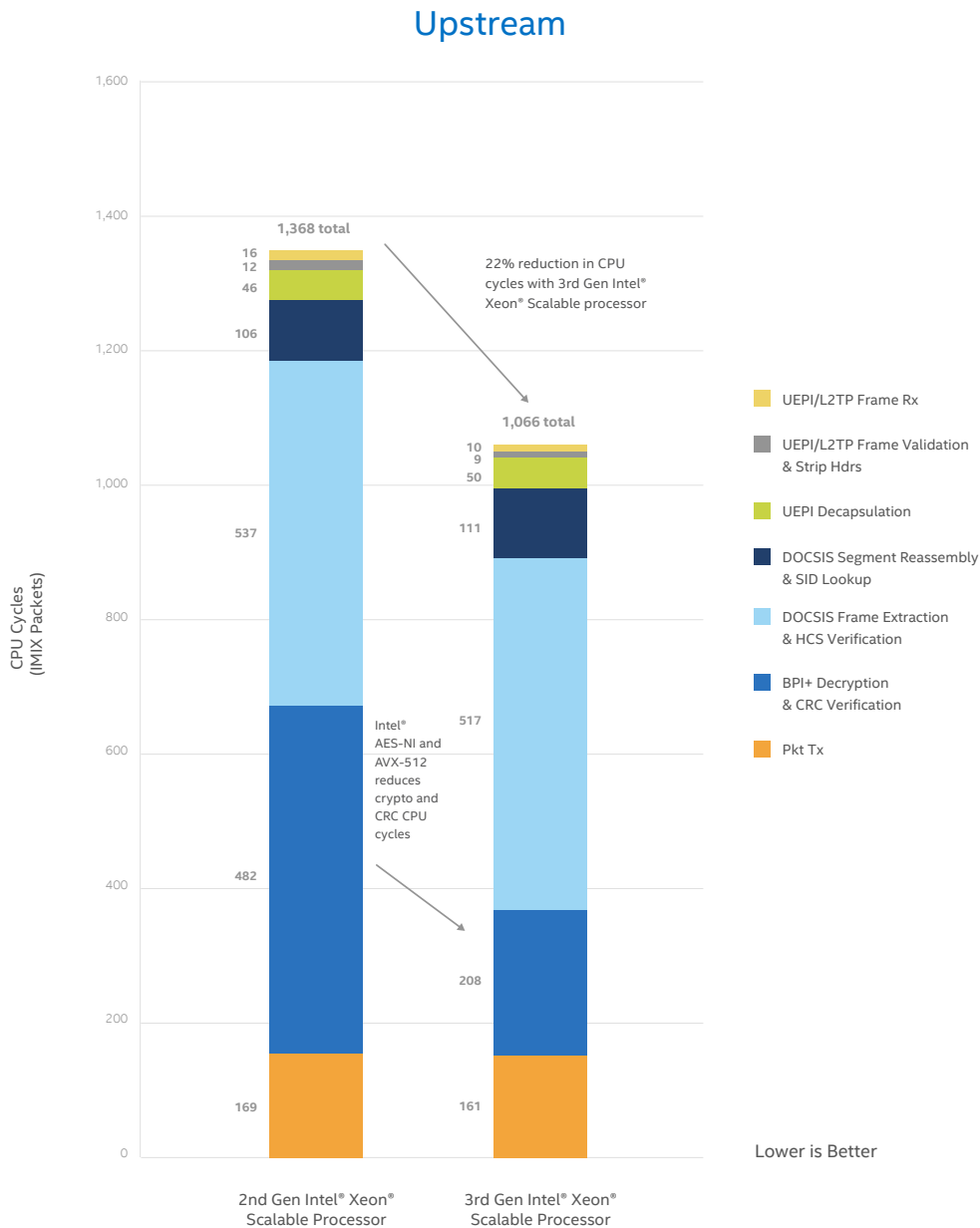


Figure 7. Upstream Data Plane CPU Cycle Comparison (IMIX)

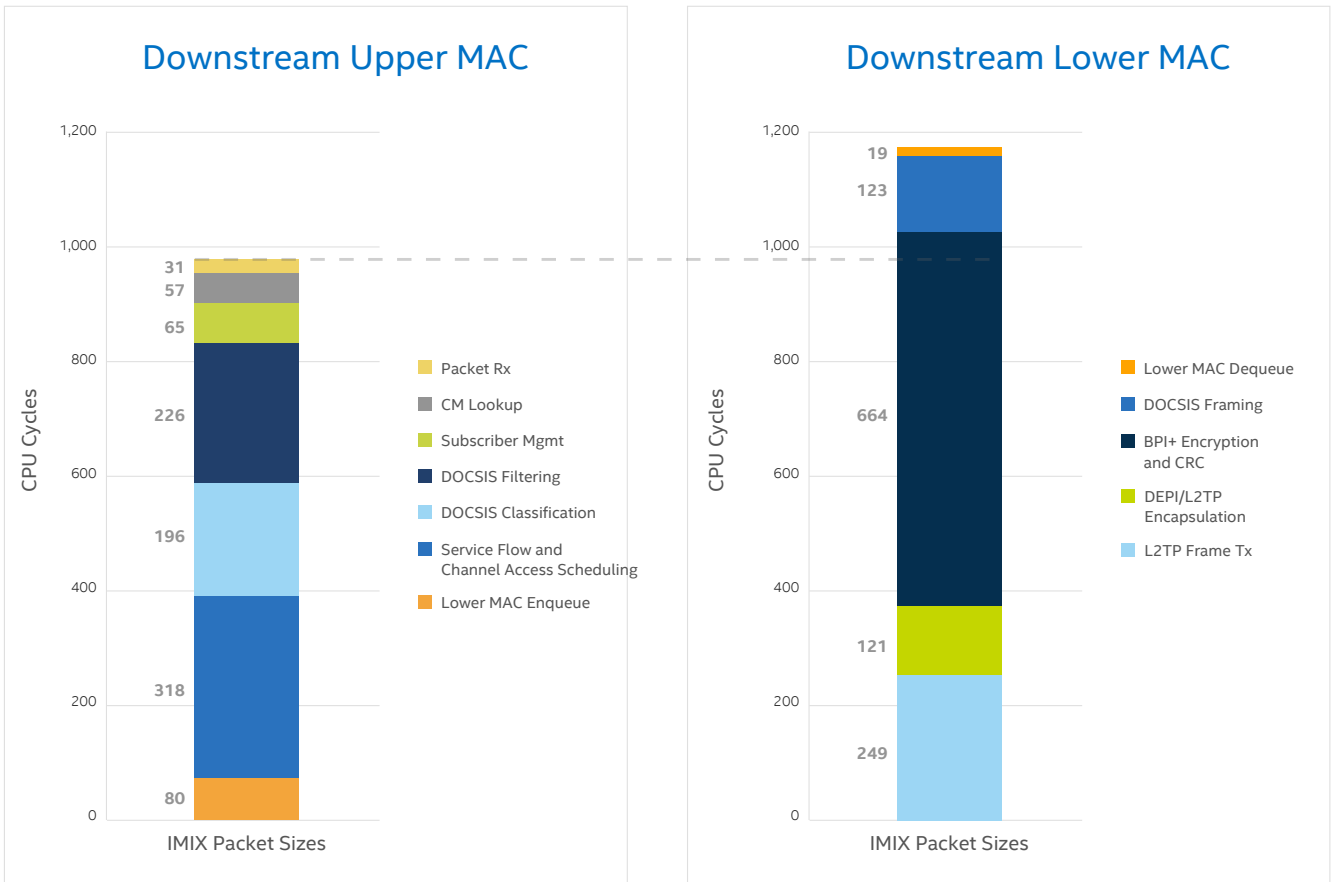


Figure 8. Downstream CPU Cycles - Upper and Lower MAC Comparison

### System Scalability and Server Sizing

For cable system architects who wish to size server requirements, scalability of the cable workload performance across CPU cores is very important. Figure 9 shows DOCSIS 3.1 (6xOFDM with AES encryption) bi-directional throughput as it scales quite linearly with the number of service groups (SGs) when passing IMIX traffic.<sup>9</sup> This is achieved because the data plane workload for each SG runs independently of each other on dedicated cores or hyper-threads with optimal sharing of key CPU resources such as L3 cache.

Downstream traffic (IMIX) is processed on a single core per SG with two upstream traffic instances per core. For this benchmark, all processing resides on just one of the two processors of a dual processor platform.

In the near-term, the bandwidth requirements for DOCSIS traffic per SG may be less. Figure 10 shows performance scalability for a channel configuration scenario with 32 legacy channels (single-carrier, quadrature amplitude modulation SC-QAM, DOCSIS v3.0 or earlier) and two OFDM channels, which has a theoretical downstream bandwidth limit per SG of approximately 5.1 Gbps.<sup>9</sup>

In this case upstream and downstream SG processing is executed on the same software thread and two of these run on sibling hyper-threads of a single processor core. Effectively SG data plane processing uses a third of the CPU core resources of the previous configuration which was for max SG performance.

Figure 11 shows a performance comparison between Intel 2nd Gen and 3rd Gen N-SKU (Networking SKU) platforms. The gen-to-gen performance gain for the maximum core-count Intel Xeon N-SKU processor is 70%.<sup>9</sup>

This sizeable performance boost is due to CPU architecture enhancements on 3rd Gen Intel Xeon processors and the extra 8 CPU cores which can support 4 extra SGs.

Note also that even higher platform throughput is possible with QAT Acceleration and Intel® Speed Select Technology. The benefits of QAT have already been covered in earlier sections. The Intel Speed Select Technology Base Frequency feature allows a select number of cores to be configured to run faster than the processor’s default base frequency, making it possible to balance power conservation and performance to best suit a vCMTS deployment.

Note also that when dimensioning a system, other processing should also be considered, which was not included in the performance data presented in this paper. For instance, the upstream scheduler, control plane, high availability (HA) standby instances, and other functionality hosted on the platform will consume additional processor cores. For the performance data presented in this paper, CPU cores have been reserved for this processing.



### vCMTS Scaling on 3rd Gen Intel® Xeon® Scalable Processors - Max B/W per Service-Group

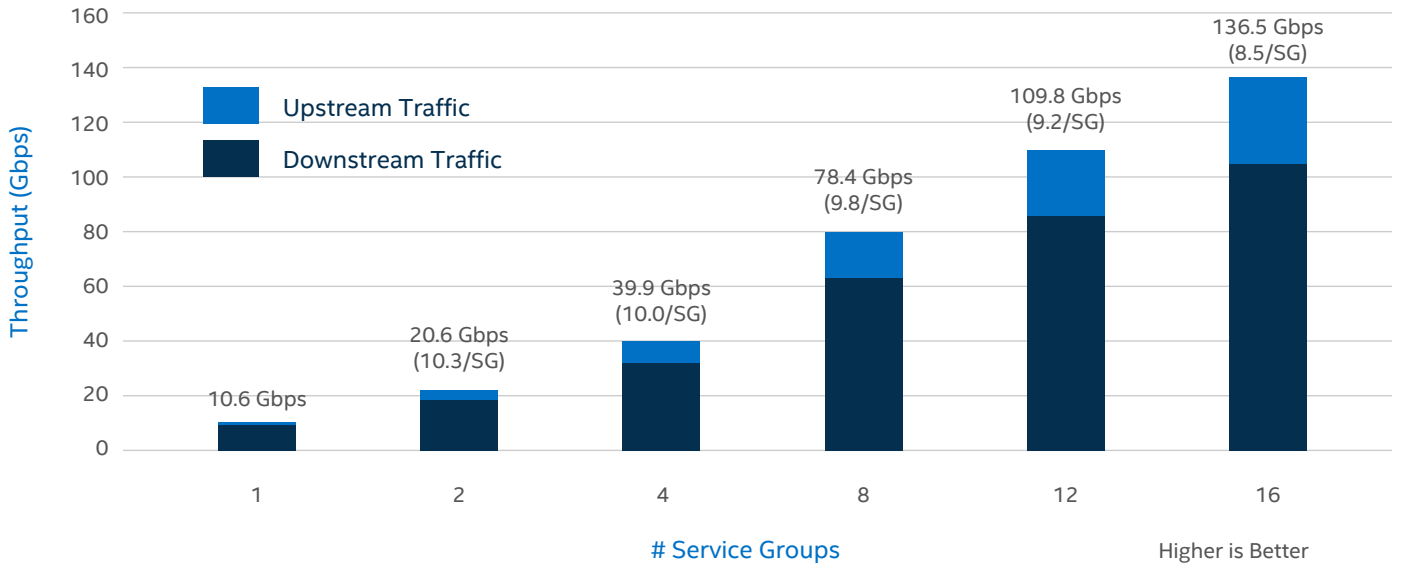


Figure 9. Platform Throughput Scalability with Maximum Bi-directional Bandwidth per Service-Group (single CPU)

### vCMTS Scaling on 3rd Gen Intel® Xeon® Scalable Processors - 2 Service-Groups per CPU Core

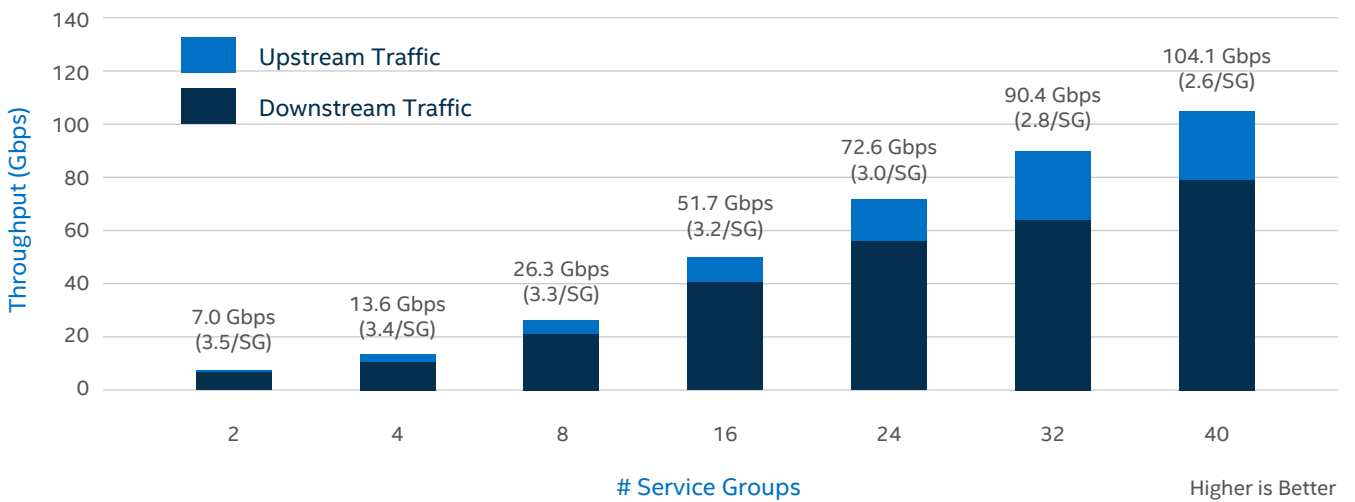


Figure 10. Platform Throughput Scalability with 2 Service-Groups per Core (single CPU)



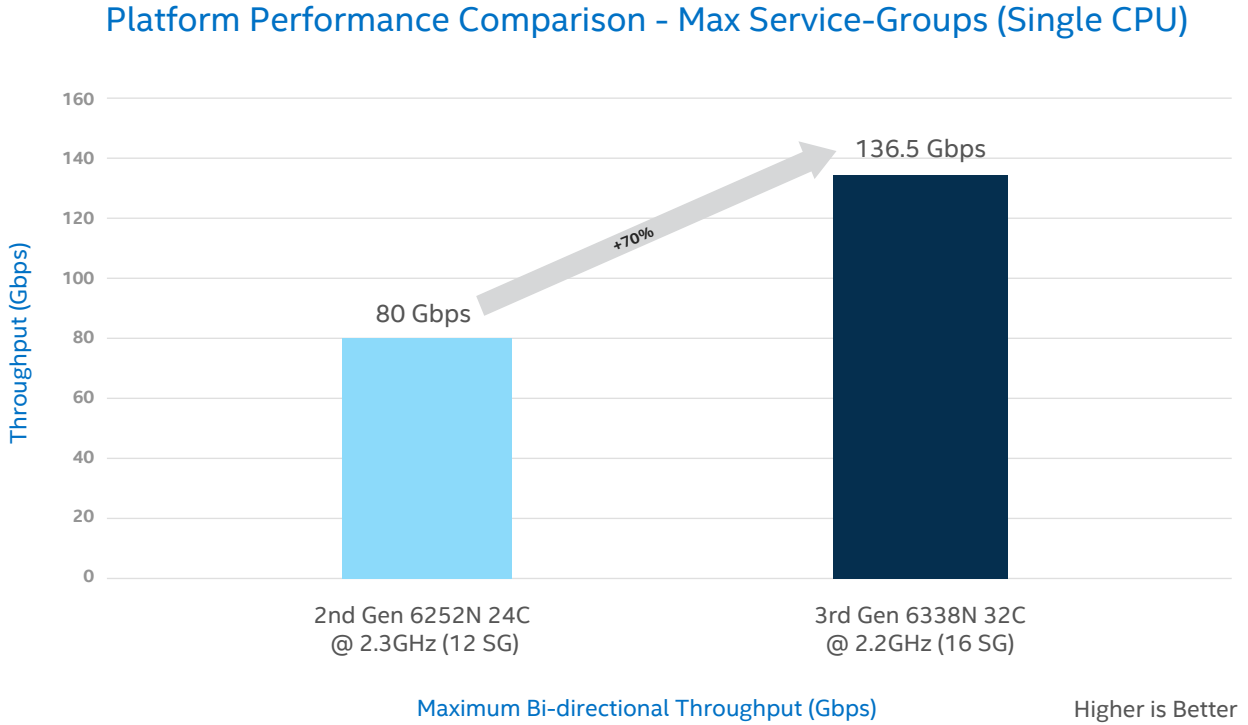


Figure 11. Platform Performance comparison for 2nd Gen and 3rd Gen Intel® Xeon® Scalable processors

### Compelling vCMTS Performance on 3rd Gen Intel® Xeon® Scalable Processors

More and more network operators are adopting software-based solutions running on industry-standard, high-volume servers to increase agility, flexibility, and cost competitiveness, while matching the performance of custom-built proprietary solutions. Utilizing the 3rd Gen Intel Xeon Scalable processor architecture offers an even higher level of performance and scalability than previous processor generations.

This paper has demonstrated how the scalability of Intel Xeon processor-based platforms can be applied to future OFDM-focused deployments, supporting close to 5 OFDM channels per CPU core in software. The flexibility to support a more near-term configuration with a mix of OFDM and

SC-QAM channels was also demonstrated. In this case a larger number of service groups may be supported, with each allocated a lower bandwidth.

Results shown in this paper showcase a 70% improvement in vCMTS platform performance and service group density when utilizing the maximum core Network SKU of the 3rd generation Intel Xeon Scalable processor, with further performance gains possible by using Intel QuickAssist and Speed Select Technologies.

With this data, network architects can better assess the benefits and costs of implementing a vCMTS on industry-standard, high-volume servers based on 3rd generation Intel Xeon Scalable processors.

For more information on networking solutions using Intel technologies, please visit [networkbuilders.intel.com](https://networkbuilders.intel.com).

## Appendix A: Intel vCMTS Reference Data Plane Packet-processing Pipeline Stages

The following describes the packet-processing stages of the Intel vCMTS Reference Data Plane Upstream and Downstream pipelines as shown in Figure 3.

### Downstream Data Plane Pipeline Stages

#### 1. Receive IP Frames

Using the DPDK Ethdev API, IP packet bursts are received via the DPDK Poll Mode Driver (PMD) from the Rx queue of a NIC VF (virtual function) port. These packets are read by a data plane software thread which begins vCMTS downstream packet processing. Packets are steered to a service group specific VF based on destination MAC address.

#### 2. Cable Modem Lookup & Subscriber Management

The DPDK Hash API is used to do a bulk lookup (i.e. with multiple packets) based on the destination IP address of the received frames to retrieve cable-modem records containing MAC address, DOCSIS filter, DOCSIS classifier, service flow queue and security info. The Destination MAC address of the Ethernet frame is also updated to a cable-modem-specific address in this stage. The number of active subscriber IP addresses is checked against the DOCSIS limit (tracked by a destination IP address list per cable-modem).

#### 3. DOCSIS Filtering

The DPDK Access Control List (ACL) API is used to apply an ordered list of DOCSIS filter rules to Ethernet frames. DOCSIS filter rule configuration is described in Appendix C.

#### 4. DOCSIS Classification

The DPDK Access Control List (ACL) library is used to apply an ordered list of rules to classify Ethernet frames for enqueueing to cable-modem service-flow scheduler queues. DOCSIS service-flow scheduler rule configuration is described in Appendix C.

#### 5. DOCSIS QoS - Service Flow & Channel Access Scheduling

The DPDK HQoS Scheduler API is used to apply rate-shaping, congestion control, and weighted-round-robin (WRR) scheduling to cable-modem service flow queues.

The DPDK Scheduler API has also been adapted to perform channel access scheduling on data packets after service-flow scheduling. Channel access scheduling is optimized by performing it in an earlier pipeline stage than is typically done in other implementations. This scheduling stage takes into account the DEPI and DOCSIS encapsulation overhead added later in the pipeline.

#### 6. Lower MAC Interface

A DPDK ring is used to transfer packets between upper MAC and lower MAC processing. This allows upper and lower MAC processing to be executed on separate threads.

#### 7. DOCSIS Framing

DOCSIS MAC headers are generated, including DOCSIS Header Check Sequence (HCS), for prepending to packets. The DPDK CRC API is used to generate the DOCSIS HCS. AVX-512 instructions are used for optimum performance on Intel 3rd Gen Xeon Scalable Processor platforms.

#### 8. IP Frame CRC Generation and DOCSIS BPI+ Encryption

The 32-bit Ethernet cyclic redundancy code (CRC) of the packet is generated and DOCSIS BPI+ encryption is then applied. These two stages are performed using DPDK combined Crypto-CRC processing. Intel Vector AES-NI, Vector PCLMULQDQ and AVX-512 instructions are used for optimum performance on Intel 3rd Gen Xeon Scalable Processor platforms.

#### 9. DEPI Encapsulation

DEPI encapsulation is performed based on the DOCSIS 3.1 specification. Frames are converted to Packet Streaming Protocol (PSP) segments, concatenated using DPDK Mbuf chaining, and encapsulated into L2TP frames of maximum transmission unit size. PSP segments are fragmented across DEPI frames so all transmitted frames are of maximum transmission unit (MTU) size in order to ensure maximum utilization of the R-PHY link.

#### 10. Transmit DEPI/L2TP Frames

Using the DPDK Ethdev API, bursts of DEPI/L2TP frames are transmitted via the DPDK Poll Mode Driver (PMD) to the NIC VF Tx queue of the associated service group.

## Upstream Data Plane Pipeline Stages

### 1. Receive UEPI/L2TP Frames

Using the DPDK Ethdev API, bursts of L2TP/IP frames containing UEPI encapsulated DOCSIS streams are received via the DPDK Poll Mode Driver (PMD) from the Rx queue of a NIC VF (virtual function) port. These frames are read by a data plane software thread which begins vCMTS upstream packet processing. Frames are steered to the service group specific VF based on destination MAC address.

### 2. Validate Frame and Strip IP Headers

The L2TP/IP frame is validated and IP headers are stripped.

### 3. UEPI Decapsulation

UEPI decapsulation is performed based on the DOCSIS 3.1 specification. UEPI/PSP sequence numbers are verified to be in order.

### 4. DOCSIS Segment Reassembly & Service ID Lookup

UEPI PSP header, data and trailer segments are traversed and the data segments are reassembled into DOCSIS stream segments. The DPDK Hash API is used to perform lookups based on service ID values to retrieve cable-modem security info.

### 5. DOCSIS Frame Extraction

DOCSIS frames are extracted from DOCSIS stream segments, including reassembly of fragmented frames using the DPDK Mbuf API.

### 6. DOCSIS Frame HCS Verification

Header Check Sequence (HCS) verification is performed for the extracted DOCSIS frames using the DPDK CRC API.

### 7. DOCSIS BPI+ Decryption and IP Frame CRC Verification

DOCSIS BPI+ decryption is applied to DOCSIS frames for AES or DES encrypted frames and the 32-bit Ethernet cyclic redundancy code (CRC) of the resulting Ethernet packet is verified. These two stages are performed using the DPDK combined Crypto-CRC processing. Intel Vector AES-NI, Vector PCLMULQDQ and AVX-512 instructions are used for optimum performance on Intel 3rd Gen Xeon Scalable Processor platforms.

Note that CRC verification is generally not required for upstream encapsulated packets so it is disabled by default.

### 8. Transmit IP Frames

Using the DPDK Ethdev API, bursts of IP frames are transmitted via the DPDK Poll Mode Driver (PMD) to the NIC VF Tx queue of the associated service group.

### Appendix B: Performance Test Environment

The performance test environment for the Intel vCMTS Reference Data Plane consists of a vCMTS platform and a software-based traffic generator platform, as shown in Figure 12.

The vCMTS platform is based on a server blade with dual 3rd Gen Intel Xeon Scalable processors, 100G Intel® Ethernet 800 Series NICs, and optional Intel QAT cards.

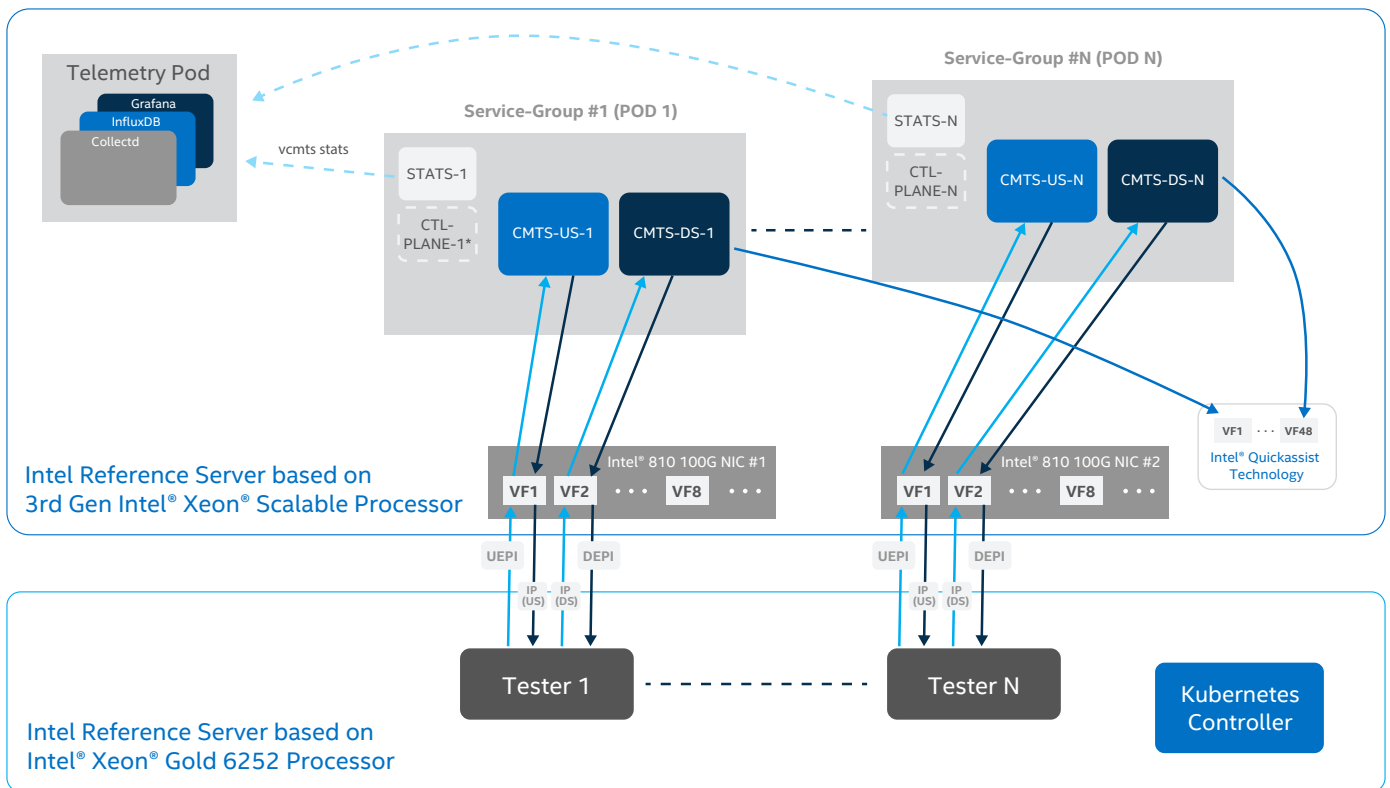
Servers based on other Intel Xeon Scalable processors or Intel® Xeon® D processors with different core counts are also supported. Different types of Intel Ethernet NICs may also be used, and the system can be configured with or without Intel QAT cards.

The entire system is deployed under a Kubernetes-orchestrated environment, in which multiple Docker containers host DPDK-based DOCSIS MAC upstream and

downstream data plane processing for individual cable SGs on the vCMTS data plane node. Intel-developed Kubernetes plugins perform resource management functions such as processor core management and assignment of single root I/O virtualization (SR-IOV) interfaces for NICs and Intel QAT devices.

Docker containers, running on the vCMTS traffic-generation node, host DPDK Pktgen-based traffic tester instances that simulate traffic into corresponding vCMTS data plane instances.

The appropriate number of NIC's and CPU cores may be used to achieve the required number of vCMTS service groups.



\*NOTE: dummy SW thread used for CTL-PLANE.

Figure 12. Performance Test Environment Used to Measure vCMTS Performance

## Appendix C: Test Environment Configuration Information and Relevant Variables

CM Lookup & Subscriber Mgmt	300 subscribers per service group, 4 IP addresses per subscriber
DOCSIS Filtering	6 filter groups, 2 filter groups associated with each cable-modem 16 filter rules per filter group 10% matched, 90% unmatched (default action – permit)
DOCSIS Classification	16 rules per subscriber, 10% matched - enqueue to one of 3 service-flow queues 90% unmatched - enqueue to default service-flow queue
Downstream Service-Flow Scheduling	8 service-flow queues per subscriber (4 active)
Downstream Channel Scheduling	6 x OFDM (1.89 Gbps) Channels, 2 x channel-bonding groups. Or 2 x OFDM (1.89 Gbps) and 32-SC-QAM (42.24 Mbps) Channels, 4 x channel-bonding groups NOTE: channel-bonding groups are distributed evenly across cable-modems
Upstream Bandwidth Scheduling	Upstream Scheduler not used. Upstream bandwidth pre-allocated in grants of 2KB per service ID. Bandwidth grants balanced evenly across 300 cable-modems.
Ethernet CRC	Downstream: 100% CRC re-generation Upstream: 0% CRC verification NOTE: CRC relates to inner frames
Encryption	100% AES, 0% DES
Packet IMIX Distribution	Upstream 65% : 84B, 18% : 256B, 17% : 1280B Downstream 15% : 84B, 10% : 256B, 75% : 1280B

## Appendix D: System Configuration

vCMTS Server - based on 3rd Gen Intel® Xeon® Scalable Processor	
<b>Hardware</b>	
Platform	Intel® Customer Reference Board (Coyote Pass)
CPU	Intel® Xeon® Scalable Gold 6338N Processor, 2.2 GHz, 32 Cores Uncore Frequency: 1.5 GHz Microcode : 0x261 NOTE: Single CPU (of dual CPU package) used for all vCMTS performance benchmarks
Memory	16 x 8GB DDR4-2667
Hard Drive	Intel® SSD (480G)
Network Interface Card	2 x Intel® Ethernet Converged Network Adapter 810 100GbE (per CPU)
Crypto Acceleration Card	1 x Intel® QuickAssist Technology Adapter 8970 (per CPU)
<b>Software</b>	
Host OS	Ubuntu 20.04, Linux Kernel v5.4.x
Container Orchestration	Kubernetes v1.18 (CMK 1.4.1, SR-IOV Plugin 3.2.0, QAT Plugin 0.11)
Linux Container	Docker v19.03
DPDK	DPDK v20.08
vCMTS	Intel vCMTS Reference Data plane v20.10
<b>Date Tested: March 11th, 2021</b>	

vCMTS Traffic Generator	
<b>Hardware</b>	
Platform	Intel® Customer Reference Board (Wolf Pass)
CPU	Intel® Xeon® Scalable Gold 6252 Processor, 2.1 GHz, 24 Cores
Memory	12 x 8GB DDR4-2993
Hard Drive	Intel® SSD (480G)
Network Interface Card	2 x Intel® Ethernet Converged Network Adapter 810 100GbE
<b>Software</b>	
Host OS	Ubuntu 20.04, Linux Kernel v5.4.x
Container Orchestration	Kubernetes v1.18 (CMK 1.4.1, SR-IOV Plugin 3.2.0, QAT Plugin 0.11)
Linux Container	Docker v19.03
DPDK	DPDK v20.08
Traffic Generator	DPDK Pktgen v19.10

The following vCMTS server configuration was used for performance comparison.

vCMTS Server - based on 2nd Gen Intel® Xeon® Scalable Processor	
<b>Hardware</b>	
Platform	Supermicro X11DPG-QT
CPU	Intel® Xeon® Scalable Gold 6252 Processor, 2.3 GHz, 24 Cores Uncore Frequency: 2.4 GHz Microcode: 0x5002f01 NOTE: Single CPU (of dual CPU package) used for all vCMTS performance benchmarks
Memory	12 x 8GB DDR4-2993
Hard Drive	Intel SSD (480G)
Network Interface Card	NOTE: Single CPU (of dual CPU package) used for all vCMTS performance benchmarks
Crypto Acceleration Card	1 x Intel® QuickAssist Technology Adapter 8970 (per CPU)
<b>Software</b>	
Host OS	Ubuntu 20.04, Linux Kernel v5.4.x
Container Orchestration	Kubernetes v1.18 (CMK 1.4.1, SR-IOV Plugin 3.2.0, QAT Plugin 0.11)
Linux Container	Docker v19.03
DPDK	DPDK v20.08
vCMTS	Intel vCMTS Reference Data Plane v20.10
<b>Date Tested: March 2nd, 2021</b>	

## Appendix E: Acronyms and Definitions

Term	Description
<b>AES</b>	Advanced Encryption Standard
<b>BPI</b>	Baseline Privacy Interface
<b>CM</b>	Cable modem
<b>CMTS</b>	Cable modem termination system
<b>CRC</b>	Cyclic redundancy code
<b>DAA</b>	Distributed access architecture
<b>DEPI</b>	Downstream External PHY Interface
<b>DES</b>	Data Encryption Standard
<b>DOCSIS</b>	Data over Cable Service Interface Specification
<b>FMA</b>	Flexible MAC architecture
<b>HA</b>	High Availability
<b>HCS</b>	Header Check Sequence
<b>MAC</b>	Media Access Control
<b>MSO</b>	Multiple system operator
<b>NFV</b>	Network Functions virtualization
<b>PSP</b>	Packet Streaming Protocol
<b>RMC</b>	Remote-MAC core
<b>RMD</b>	Remote-MAC-PHY device
<b>R-PHY</b>	Remote Phy
<b>SG</b>	Service group
<b>UEPI</b>	Upstream External Phy Interface
<b>vCore</b>	Virtualized core
<b>vCMTS</b>	Virtualized cable modem termination system





## References

1. Intel vCMTS Reference Data Plane : <https://01.org/access-network-dataplanes>
2. Intel Multi-Buffer Crypto Library : <https://github.com/intel/intel-ipsec-mb>
3. Data Plane Development Kit (DPDK) : <https://www.dpdk.org>

## Footnotes

1. 50% more L1 cache based on 36KB per core on 2nd Gen Intel® Xeon® Scalable processor and 48KB per core on 3rd Gen Intel® Xeon® Scalable processor
2. 25% more L2 cache based on 1MB per core on 2nd Gen Intel® Xeon® Scalable processor and 1.25MB per core on 3rd Gen Intel® Xeon® Scalable processor
3. One AES-NI port per core for Crypto on 2nd Gen Intel® Xeon® Scalable processor and Two per core on 3rd Gen Intel® Xeon® Scalable processor
4. 42% more CPU cores based on max core-count of 28 cores per CPU on 2nd Gen Intel® Xeon® Scalable processor and 40 cores per CPU on 3rd Gen Intel® Xeon® Scalable processor
5. 45% more Memory B/W based on 6 memory channels per CPU up to 2933 MT/s on 2nd Gen Intel® Xeon® Scalable processor and 8 channels up to 3200 MT/s per CPU on 3rd Gen Intel® Xeon® Scalable processor
6. 33% more PCI Lanes based on 3 x 16 Lanes per CPU on 2nd Gen Intel® Xeon® Scalable processor and 4 x 16 Lanes per CPU on 3rd Gen Intel® Xeon® Scalable processor
7. 2 x PCI B/W based on PCIe Gen3 on 2nd Gen Intel® Xeon® Scalable processor and PCIe Gen4 on 3rd Gen Intel® Xeon® Scalable processor
8. 11% more L3 cache per core based on 1.375MB per core on 2nd Gen Intel® Xeon® Scalable processor and 1.5MB per core on 3rd Gen Intel® Xeon® Scalable processor
9. Performance measured using the system configuration described in Appendix B, C and D. Note that there will be a margin of error due to the action of taking performance measurements. Results shown are for a reference implementation of a vCMTS data-plane and not a production system. These numbers should be treated strictly as a reference only.
10. Hyper-threaded siblings are hardware threads of execution contained within the same physical CPU core and which share the same set of core resources. For data-plane cores on the Intel reference vCMTS system, each hyper-thread runs its own data-plane software thread.

## Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more at [www.Intel.com/PerformanceIndex](http://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.

Your costs and results may vary.

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

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.

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.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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

© 2021 Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.