White Paper

Dynamic Power Savings in Cloud-Native 5G Wireless Infrastructure Network Functions

Authors

SK Telecom ICT Infra Tech, Infra Strategy & Tech Center

DongJin Lee (dongjin@sk.com)
B5G/6G Core R&D Architect

HyungDu Choi
(hyungdu.choi@sk.com)
5G Core R&D Architect

SeongJun Lee
(seoul.lee@sk.com)
5G Core R&D Architect

JoongGunn Park
(clark.park@sk.com)
Core Architect

Intel Corporation

Chetan Hiremath
(chetan.hiremath@intel.com)
Senior Principal Engineer

Andriy Glustsov
(andriy.glustsov@intel.com)
System Software Development Engineer

Vishal Deep Ajmera
(vishal.ajmera@intel.com)
System Software Architect

Kashish Keshavkumar Singh
(kashish.singh@intel.com)
Software Engineer

Gordon Noonan
(gordon.noonan@intel.com)
Software Engineer

Shakun Yawatkar
(shakun.rajan.yawatkar@intel.com)
Software Engineer

1. Abstract

This paper describes techniques for improving energy efficiency in 5G wireless core infrastructure for the purpose of reducing energy costs incurred by Communication Service Providers (COSP) and furthering sustainability goals by reducing overall greenhouse gas emissions.

Intel and SK Telecom have developed a new dynamic power saving mechanism for wireless packet core networks which adjusts CPU frequencies in real time in response to packet burst and utilization telemetry. By reacting to the continuously varying nature of traffic in real time, energy savings can be realized at finer granularity.

Using SK Telecom's traffic models from the 5G commercial network for user plane packet processing workload (5G UPF), we have analyzed how CPU frequencies can be adjusted dynamically without introducing packet error or loss. Our prototype demonstrated power reduction by up to 55% during the non-busy hour and 30% during the busy hour, resulting in a 42% reduction over the 24-hour period.

Similarly, using SK Telecom's traffic models for control plane showed that power consumption can be reduced by up to 42% during the non-busy hour and 19% during the busy hour, for a total reduction of 34% over a 24-hour period.

2. Background and Introduction

Historically, wireless infrastructure has focused on transitions from proprietary appliance architecture-based implementations to Network Function Virtualization (NFV) based deployments on Intel® Xeon® Processor-based platforms, leveraging technologies such as virtualization for processor and memory, and SRIOV for I/O virtualization.

As such, most 4G/LTE and current 5G deployments focused entirely on performance, density and capacity scaling.

Network architecture continues to evolve with deployments further out in the network, such as at the network edge and central offices. This evolution away from traditional telco data centers increases the breadth of deployment to reach more customers at higher bandwidth, while reducing latency by offloading traffic to application servers closer to the edge.

Coupled with this are exponential increases in energy costs incurred by CoSPs and the critical need to address sustainability goals by cutting overall greenhouse gas emissions and saving social costs.

5G wireless core infrastructure comprises a range of network functions (NFs) as illustrated in Figure 1. These network functions are deployed across physical locations that range from traditional central telecommunications data centers, regional central offices and localized edge data centers with local breakouts to data networks where edge applications may be deployed as illustrated in the diagram in Figure 2. Edge deployments reduce overall traffic trunked to the central data centers and improve end-to-end latencies for applications.

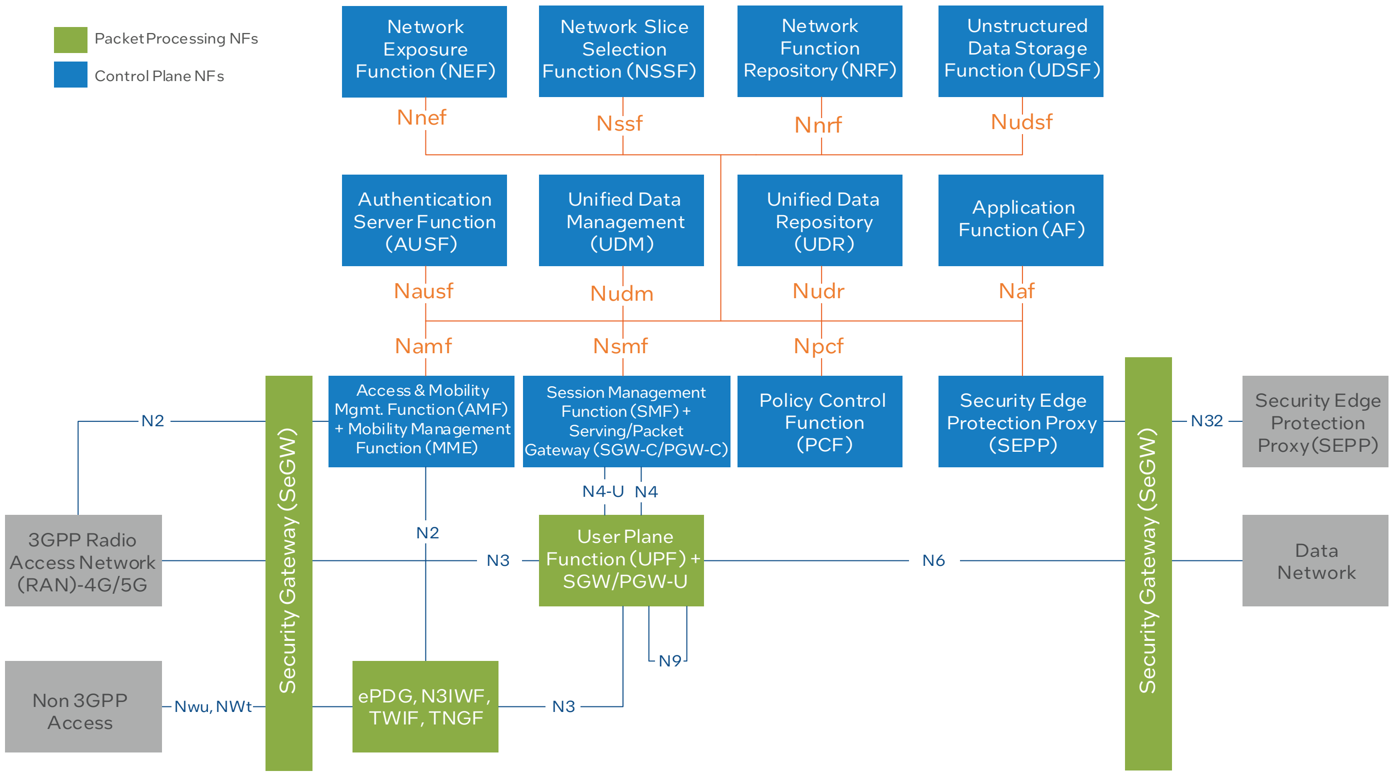

Packet Processing NFs
Control Plane NFs
Network Exposure Function (NEF)
Network Slice Selection Function (NSSF)
Network Function Repository (NRF)
Unstructured Data Storage Function (UDSF)
Nnef
Nssf
Nnrf
Nudsf
Authentication Server Function (AUSF)
Unified Data Management (UDM)
Unified Data Repository (UDR)
Application Function (AF)
Nausf
Nudm
Nudr
Naf
Namf
Nsmf
Npcf
Access & Mobility Mgmt. Function (AMF) + Mobility Management Function (MME)
Session Management Function (SMF) + Serving/Packet Gateway (SGW-C/PGW-C)
Policy Control Function (PCF)
Security Edge Protection Proxy (SEPP)
Security Gateway (SeGW)
Security Gateway (SeGW)
Security Edge Protection Proxy (SEPP)
N2
N32
N4-U N4
3GPP Radio Access Network (RAN)-4G/5G
N2
N3
User Plane Function (UPF) + SGW/PGW-U
N6
Data Network
N9
Non 3GPP Access
Nwu, NWt
ePDG, N3IWF, TWIF, TNGF
N3

Figure 1. 5G System Architecture

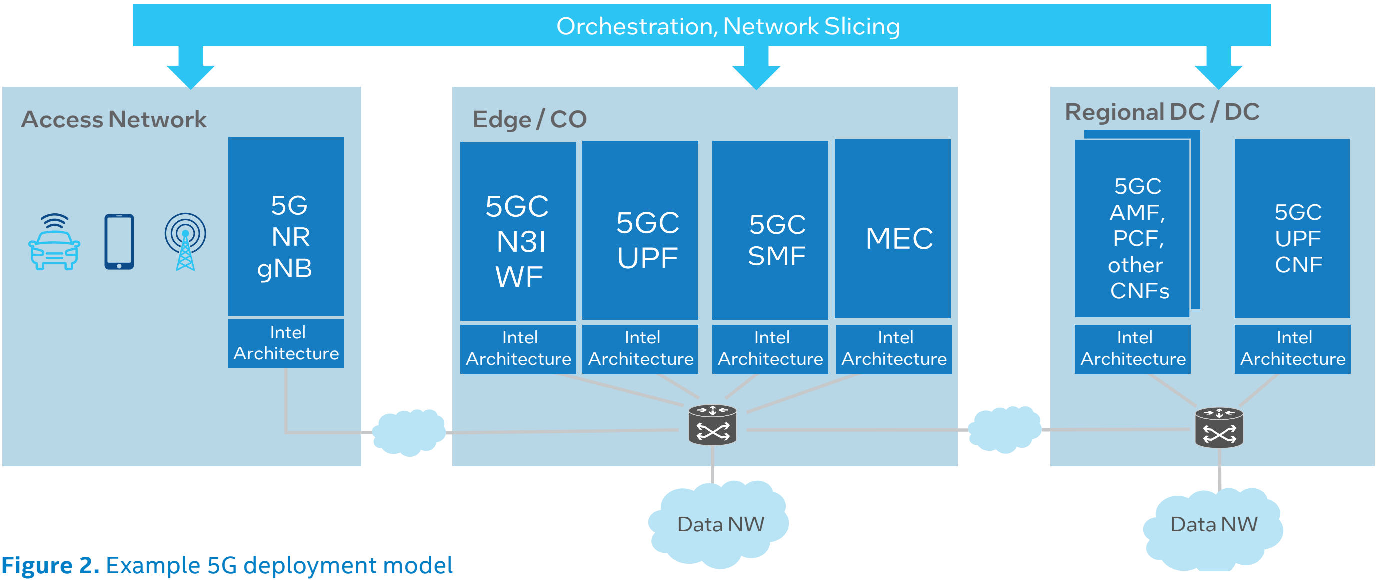


Figure 2. Example 5G deployment model

5G Network Functions are being designed, developed and deployed with cloud native technologies, with micro-services as underlying design strategy to partition network functions into a set of smaller functional entities that are loosely coupled, but tightly aligned at runtime to achieve scalability, performance, capacity, resiliency, etc. These design patterns enable CoSPs to deploy network functions across different locations using the same technologies across their network.

Figure 3 shows SK Telecom's commercial typical network throughput (CPU load) for three different sampled interfaces from the user plane over a 24-hour period. We observe the following:

- Traffic data usage behaviors are similar in general. For example, there is less traffic during the non-busy hours, and more traffic during the busy hours.
- Interfaces processing data traffic have different packet data rates depending on the traffic load (re)balancing of the user plane selection from the control plane functions (AMF/MME and SMF/GW-C).
- CPU load highly correlates with the amount of processing of the packet data rate.

In addition, we observe that existing VNF/CNF-based telco servers are typically configured for maximum performance mode. With no power saving techniques, CPU power consumption is always at maximum. Although this strategy addresses the peak requirements of packet rates, throughput, number of sessions and call attempts, this level of power consumption is excessive during non-peak times. Compounding the problem, the hours of peak requirements are different depending on the physical location of the deployment, holidays, and community events.

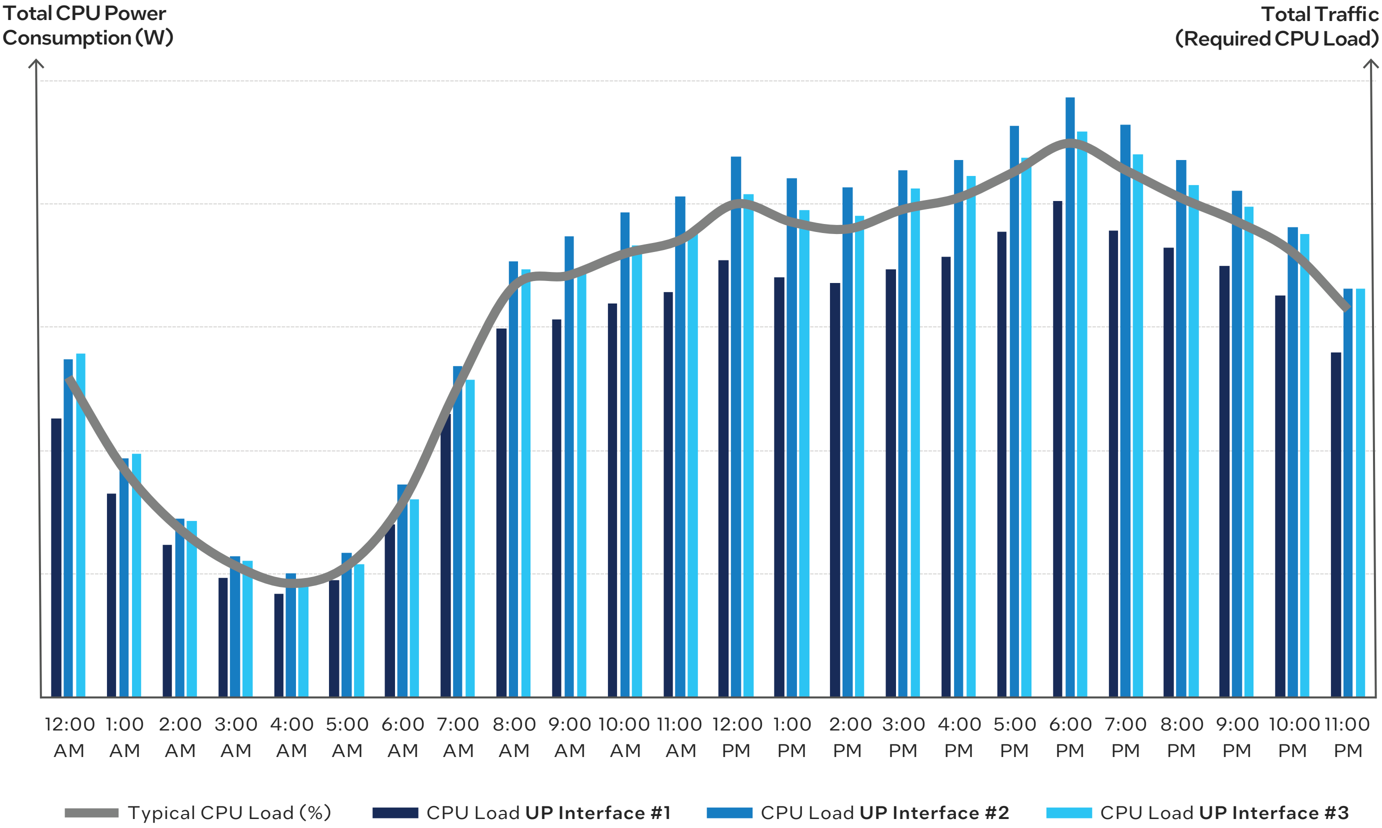


Figure 3. SK Telecom's CPU loads for different GW interfaces

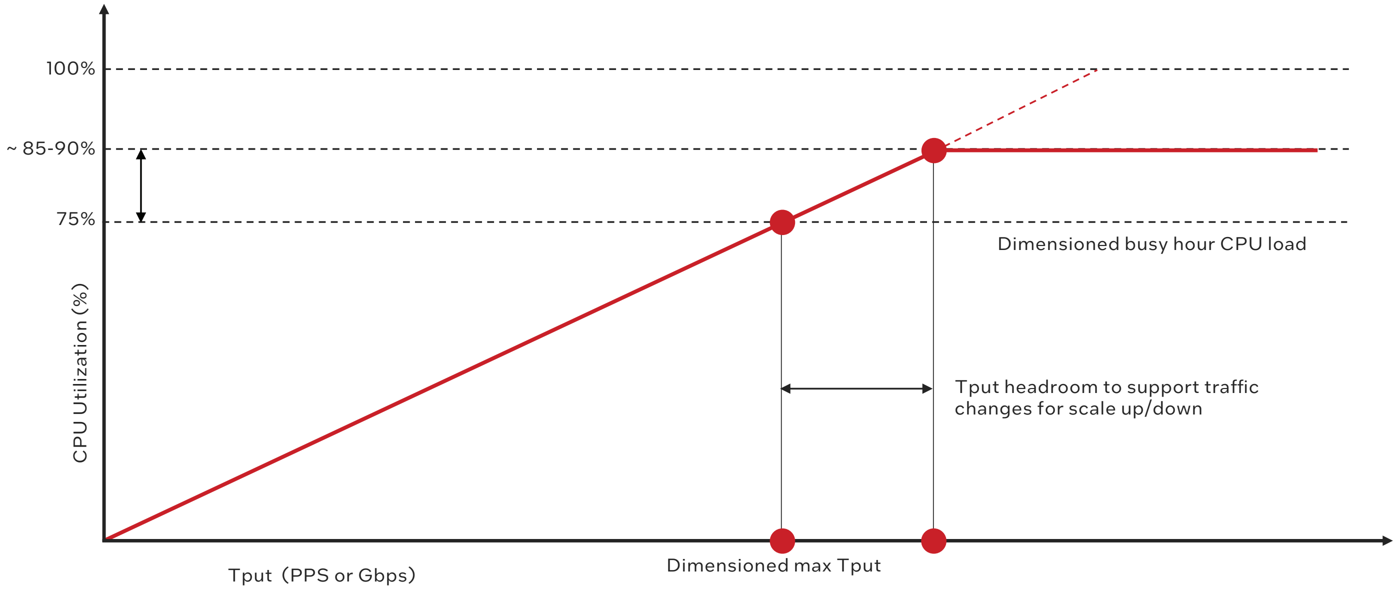


Figure 4. Typical telco system dimensioning practice

As such, the approach taken for realizing energy efficiency savings needs to be architected to be independent of specific physical locations where network functions are deployed, or requiring analyzing macro nature of the traffic, and minimize network level intelligence/intervention required to be able to maximize power savings, dynamically at runtime.

As shown in Figure 4, typical telco deployments are designed to ensure a certain level of guaranteed capacity, with sufficient headroom to support some amount of increased traffic before triggering load regulation or rejecting new sessions or transactions that would negatively affect key performance indicators (KPIs).

Power management capabilities play a critical role in improving energy efficiencies in a real world operational network environment and reduce the carbon footprint.

The power management architecture and implementation must also give TEM/ISVs and CoSPs these abilities:

- The ability to deliver key metrics for network availability and resiliency (e.g., 5-nines).
- Scalability that allows for targeting a range of applications for power management to improve overall telco data center, central office and edge data center energy efficiency.

This paper describes techniques for improving energy efficiency in 5G wireless core infrastructure for the purpose of reducing energy costs incurred by CoSPs and furthering sustainability goals by reducing overall greenhouse gas emissions.

This document focuses on techniques for power management at the cloud-native packet-processing user plane and server-level control plane, which can be expanded to cluster-level power controls. Similar techniques scaled to virtual machine VNFs will be addressed in future versions of this document.

2.1 Packet Behaviors from the Commercial Network

Figure 5 shows raw packet traces captured at user plane interfaces at different locations at SK Telecom's commercial network.

These traces reveal a scale-invariant burstiness behavior down to the millisecond level, known as "self-similarity." In other words, the number of packets and streams entering the system show a similar burstiness at any zoom level of the time axis. This is a widely known observation that has been measured and analyzed globally in general and makes it difficult to predict upcoming packet arrivals.

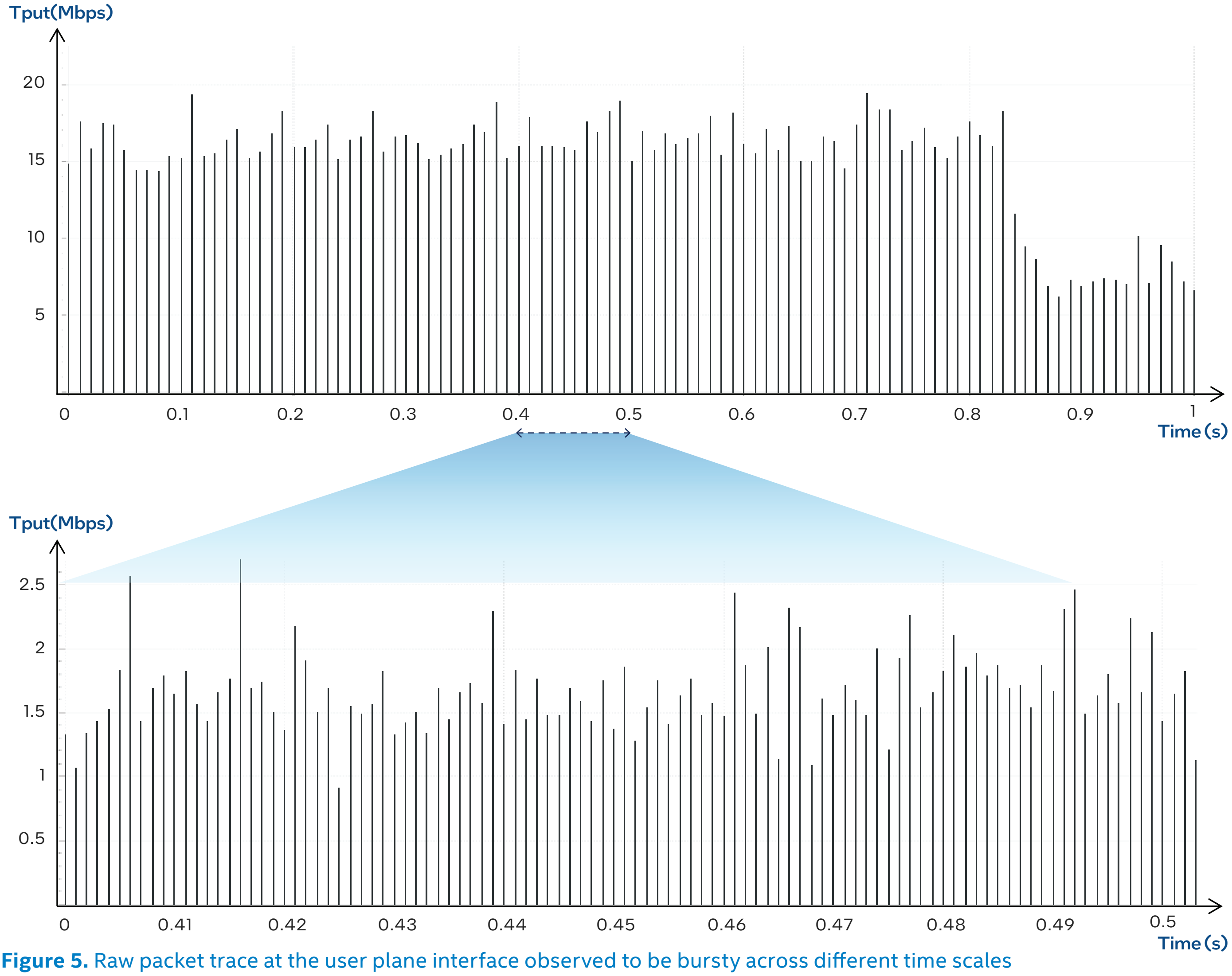


Figure 5. Raw packet trace at the user plane interface observed to be bursty across different time scales

Because the timing and duration of packet burst arrivals cannot reliably be predicted in real time, we believe that it is more important to develop a method that does the following:

- Gather, in nearly real time, the packet queue status without degrading performance.
- Adjust packet processing requirements to instantly trigger CPU resource allocations dynamically.
- Tune for favoring zero packet loss to guard against unexpected micro-burst behaviors, across all time periods including non-busy and busy times.

In summary, because packet rates and burstiness are impossible to predict, intelligent approaches to power management are required that do not adversely affect KPIs such as those relating to packet loss or latency.

3. Platform Technologies for Power Management

To realize overall energy efficiency improvements, power management techniques are required at various levels of the infrastructure:

- On-silicon support for software-controlled runtime power savings. This includes per-core frequency control such as P-states, per-core low power states such as C-states, and software- or hardware-driven autonomous core and uncore power controls.
- Server-level power control to proactively power down unused servers or transition them to lower power states (e.g., standby).
- Cluster-level power control to leverage telemetry to consolidate lightly loaded workloads, move entire servers into low power states, etc.

3.1 Key Capabilities in Intel Platforms for Power Management

This section describes power management capabilities in Intel Xeon Processor architectures. Many of these capabilities have been available but not utilized in wireless infrastructure network functions, as historically the focus has been only on performance. As such, these capabilities have been traditionally disabled on the platforms. We now have opportunities to improve energy efficiency in wireless infrastructure network functions by making use of these traditionally underutilized capabilities along with newer capabilities and additional improvements, including a reduction in transition latencies (described later in this section).

Per-Core C-States: Typically, each CPU core executes instructions in C0-state. Execution can be halted and the core can be transitioned into a lower power state, such as C1- or C6-state, for a period of time. The deeper power states, such as C6-state, have greater power savings but also longer latency when transitioning back to C0-state to resume execution. Table 1 shows the state of the CPU core clocks, caches and power consumption associated with different C-states.

The latencies shown in Table 1 apply to the 3rd Gen Intel Xeon processor. Software instructions can directly initiate a transition to C1- or C6-state, or software can provide hints to the processor. Further, on a 4th Gen Intel Xeon Processor, the MWAIT power management instruction accepts a hint and an optional extension that inform the processor that it can enter a specified C-state while waiting for an event or store operation on an address range specified by the MONITOR instruction. These instructions are executable at privilege level 0 (e.g., by a kernel-based CPU frequency scaling governor).

Core Frequency Scaling: This capability (also called core performance state, abbreviated as P-state) allows the frequency of each core in the CPU to be changed independently and dynamically at runtime. The core frequencies in Intel Xeon CPUs can be adjusted at a resolution of 100 MHz. This allows fine-tuning the frequency to match the actual workload. For example, in an Intel Xeon 6338N CPU with a rated frequency of 2.2 GHz, workloads can choose from up to 15 P-states between 800 and 2200 MHz. Enabling Turbo Boost (elaborated below) can opportunistically increase the CPU core frequencies further.

Uncore Frequency Scaling: This capability enables changing the frequency of the CPU logic that interconnects cores, L3 caches, memory and I/O controllers. The optimal uncore frequency selection can be determined by CPU hardware or by a software-driven selection. As an example, the Intel Xeon 6338N CPU supports between 800 MHz and 1600 MHz uncore frequencies, and an uncore Turbo frequency of up to 2200 MHz that can be changed at a fine granularity of 100 MHz.

The 3rd Gen Intel Xeon Processor has additional improvements that reduce the latency associated with power state transitions. These include:

- **Fast core frequency change**. This allows the core frequency to move from the current to the target frequency in a continuous sweep without stopping the clocks. This allows fast P-state transitions to optimize power versus performance without the latency cost.
- **Coherent fabric ("mesh") drainless frequency change**. This allows the uncore mesh PLL to transition from the current to the target frequency without draining the buffers, reducing the frequency transition time by about 3X.

Table 1. C0, C1, and C6 power states

Operational State	Core Clocks	Core Caches	Shared Caches	Core Power
C0	ON	ON	ON	██████
C1	OFF	ON	ON	████
C6	OFF	OFF	Partially flushed	██

Table 2. Transition latencies

	2nd Generation Intel® Xeon® processor	3rd Generation Intel Xeon processor
Core frequency transition block time	12 μsecs	~0 μsecs
Mesh frequency transition – I/O block time	20 μsecs	7 μsecs
Typical C6-state exit time	30 μsecs	~20 μsecs

Table 2 summarizes the key improvements in power management.

These improvements directly impact key performance indicators for latency-sensitive, high throughput packet processing and signaling workloads in 5G wireless infrastructure network functions. For example, ingress-to-egress packet processing latency under high load conditions on the Intel FlexCore 5G UPF software stack is in the range of 80 to 90 microseconds, as minimal latency for frequency changes or transitions out of lower power states is highly desirable.

To further enhance user space power management, the 4th Generation Intel Xeon Processor Family (Sapphire Rapids) implements additional sub-C0 power states called C0.1 and C0.2. These states are shallower than C1 with reduced exit latencies. These power states can be exercised by user-space software through instructions called UMWAIT and UMONITOR. This allows user space software to manage CPU power consumption in response to workload, independent of the underlying operating system or privileged mode software.

Similar power states can also be leveraged by execution of a new timed pause instruction (TPAUSE) that instructs the processor to enter an implementation-dependent optimized state until the time stamp counter of the processor reaches a predetermined value. These improvements enable user space applications such as low latency, high packet rate/throughput user space packet processing applications such as the ones enumerated at the outset of this document. These capabilities can be leveraged in user-space packet processing applications by upgrading relevant DPDK libraries that use these instructions. These power management capabilities can be then enabled by invoking API callbacks in the ethdev poll mode drivers.

Package C-States: These states are automatically managed by the processor in response to the individual core power states and the status of the platform components. Utilizing this capability requires consolidation of workloads into fewer CPUs and servers, which would allow the underlying operating system to transition the CPU cores to lower power states.

Turbo Boost: Typically, CPUs are configured to operate anywhere between the lowest (Pn), and the maximum (P1) guaranteed operating frequency. The CPU can also be configured by the BIOS or software to operate at frequencies higher than P1 in an opportunistic manner, called Turbo Boost or P0 frequency. In this state, the CPU hardware automatically evaluates the load on each of the cores in the CPU along with thermal conditions on the CPU to determine an optimal turbo frequency. In this configuration, the CPU may opportunistically increase the frequency of one or more cores to a frequency higher than P1 based on load and system thermal conditions. This can significantly improve overall performance when required.

Figure 6 illustrates overall performance states.

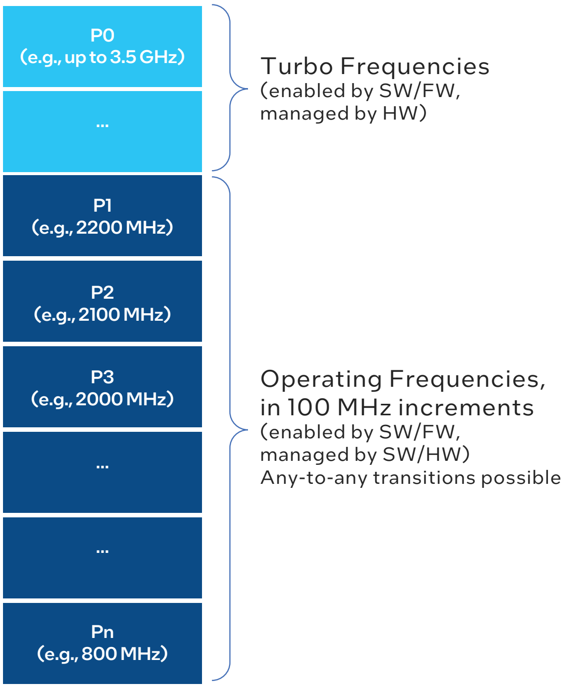

P0
(e.g., up to 3.5 GHz)
...
P1
(e.g., 2200 MHz)
P2
(e.g., 2100 MHz)
P3
(e.g., 2000 MHz)
...
...
Pn
(e.g., 800 MHz)
Turbo Frequencies
(enabled by SW/FW, managed by HW)
Operating Frequencies, in 100 MHz increments
(enabled by SW/FW, managed by SW/HW)
Any-to-any transitions possible

Figure 6. Range of P-states on Intel Xeon Processors

Table 3 below summarizes the applicability of various CPU states for dynamic, runtime power management.

Table 3. Mapping of C/P states to 5G Core workloads

CPU Power State → Workload type ↓	C0	C1	C6
User Space Packet Processing	Yes	Yes	No
Control Plane / Signaling	Yes	Yes	Yes

4. Methodology for Dynamic Power Saving in User Plane Network Functions

5G network functions can be broadly categorized into two types based on workload characteristics.

4.1 User/Data Plane Power Management

User plane function (UPF) workloads process data traffic of very high throughput (measured by packets per second or gigabits per second). These workloads are typically designed with a user space network stack and use poll mode drivers for a direct data path to network I/O (e.g., DPDK), while user space network stack is based on vector packet processing (VPP) or similar proprietary implementations from TEMs/ISVs. Examples of such workloads include User Plane Function (UPF), Central Unit (CU), Evolved Packet Data Gateway (ePDG), non-3GPP Interworking Function (N3IWF), Serving gateway (especially the SGW-U part of SGW), packet gateway (especially the PGW-U part of PGW), Broadband Network Gateway (BNG), Deep Packet Inspection (DPI), Carrier Grade Network Address Translation (CG-NAT), Firewall and Security Gateways (SeGW).

These categories of workloads typically show up as 100% utilized at the operating system level regardless of the actual packet rate or throughput that is processed or delivered. They are primarily driven by the poll mode I/O to network interface ports or to queues used to transfer packets and information between cores to enable the lowest latency and highest throughput. Furthermore, software threads that implement packet processing pipelines are typically pinned to cores that are isolated from the rest of the system threads (e.g., kernel, interrupts, timers, system daemons, etc.) to ensure that there is no interference with packet processing.

Intel and SK Telecom's joint studies have shown a correlation between computes on each core of the processor and its ability to process packets. Overall user plane power management works by analyzing utilization telemetry at a fine granularity and matching the load with an appropriate core frequency.

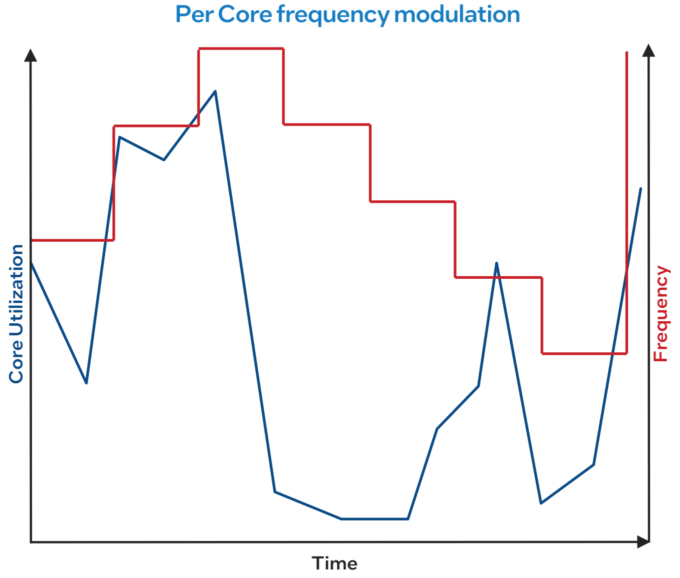


Figure 7. Near real-time tracking and adjustment of CPU core frequencies

Figure 7 conceptually illustrates tracking of core utilization and the associated frequency that can change over time. This approach of fast tracking, computation of target frequency and changes to core frequency at near real-time enables a maximum power saving approach that does not have to rely on macro level occurrences such as time of day, day of week, deployment location or unforeseen events. The algorithm chosen for computing the per-core target frequency based on runtime tracking of core utilization has an inherent bias towards maintaining key performance indicators such as packet rate or throughput.

Packet processing applications are designed and implemented either as pipelines where packets are processed in multiple stages across multiple cores, or where they run to completion a single core, as illustrated in Figures 8 and 9. As such, packets are read in from NIC ports or transferred between cores using various poll mode driver types such as ethdev, cryptodev, compressdev, rawdev, bbdev or regexdev, or with packet distribution libraries such as eventdev, distributor and ring.

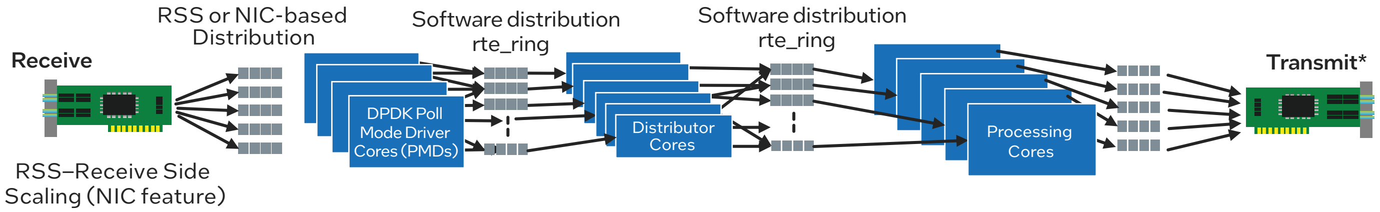

Receive
RSS or NIC-based Distribution
Software distribution rte_ring
DPDK Poll Mode Driver Cores (PMDs)
Software distribution rte_ring
Distributor Cores
Processing Cores
Transmit*
RSS–Receive Side Scaling (NIC feature)

Figure 8. Pipelined packet processing

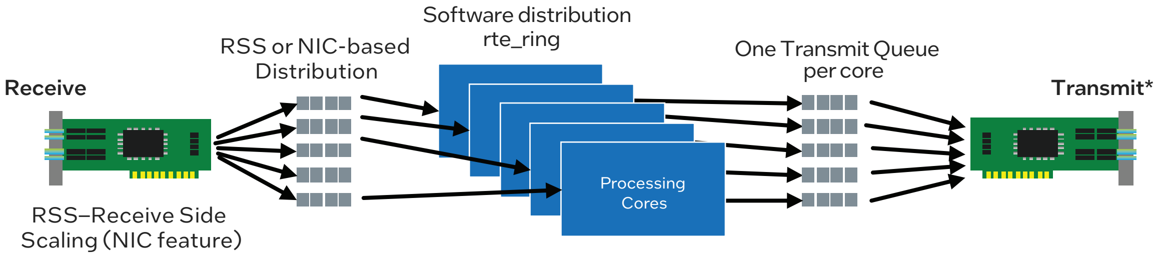

Software distribution rte_ring
RSS or NIC-based Distribution
One Transmit Queue per core
Receive
Transmit*
Processing Cores
RSS–Receive Side Scaling (NIC feature)

Figure 9. Run to complete packet processing

The compute complexity of each packet depends on various factors. Packet processing applications tend to continuously poll queues associated with network ports or they poll software queues. In either case, polled queues might be empty. We now have enhancements in the packet processing software stacks that can determine, in an application-transparent manner, the computation consumed in executing real work versus polling empty queues. Actual core utilization can be calculated and leveraged to further improve power management capabilities.

Towards that, an efficient implementation of user plane power management relies on few key principles:

- No changes are required to the Linux operating system or kernel.
- No changes are required in user space packet processing network function (e.g., 5G UPF) application logic.
- Upgrades are required in packet processing network function libraries (e.g., DPDK, VPP) that implement tracking and reporting of utilization telemetry.

An overall power management solution is implemented as a dedicated Kubernetes pod, called Intel® Infrastructure Power Manager (Intel IPM), deployed on every node in the Kubernetes cluster, as shown in Figures 10 and 11.

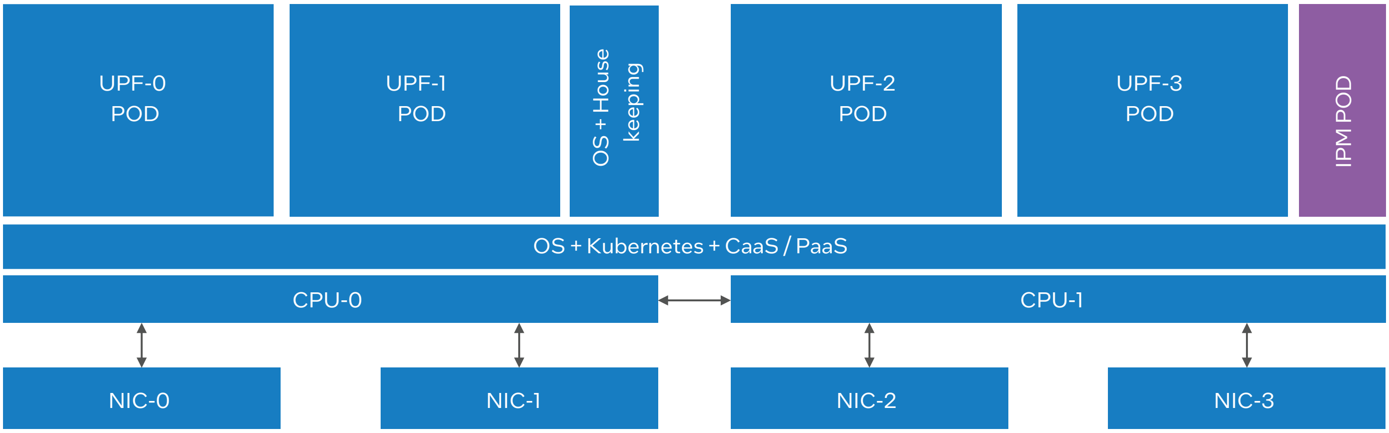


Figure 10. Example IPM deployment model on cloud native platform

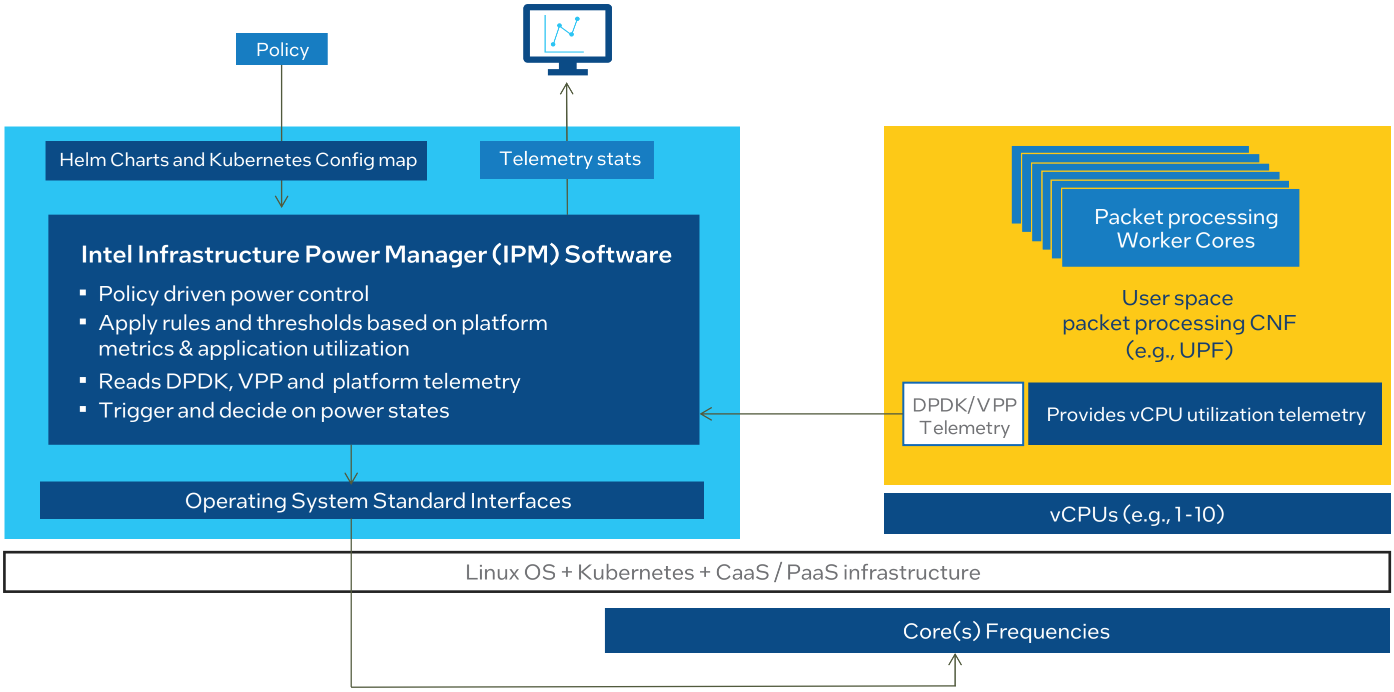

Policy
Helm Charts and Kubernetes Config map
Telemetry stats
Intel Infrastructure Power Manager (IPM) Software
▪ Policy driven power control
▪ Apply rules and thresholds based on platform metrics & application utilization
▪ Reads DPDK, VPP and platform telemetry
▪ Trigger and decide on power states
Operating System Standard Interfaces
Packet processing Worker Cores
User space packet processing CNF (e.g., UPF)
DPDK/VPP Telemetry
Provides vCPU utilization telemetry
vCPUs (e.g., 1-10)
Linux OS + Kubernetes + CaaS / PaaS infrastructure
Core(s) Frequencies

Figure 11. IPM block diagram and its interface to DPDK/VPP-based UPF Application

This IPM pod automatically detects packet processing network functions depicted here with 5G UPF pods on a given node of a Kubernetes cluster. The generic steps for power management are described below:

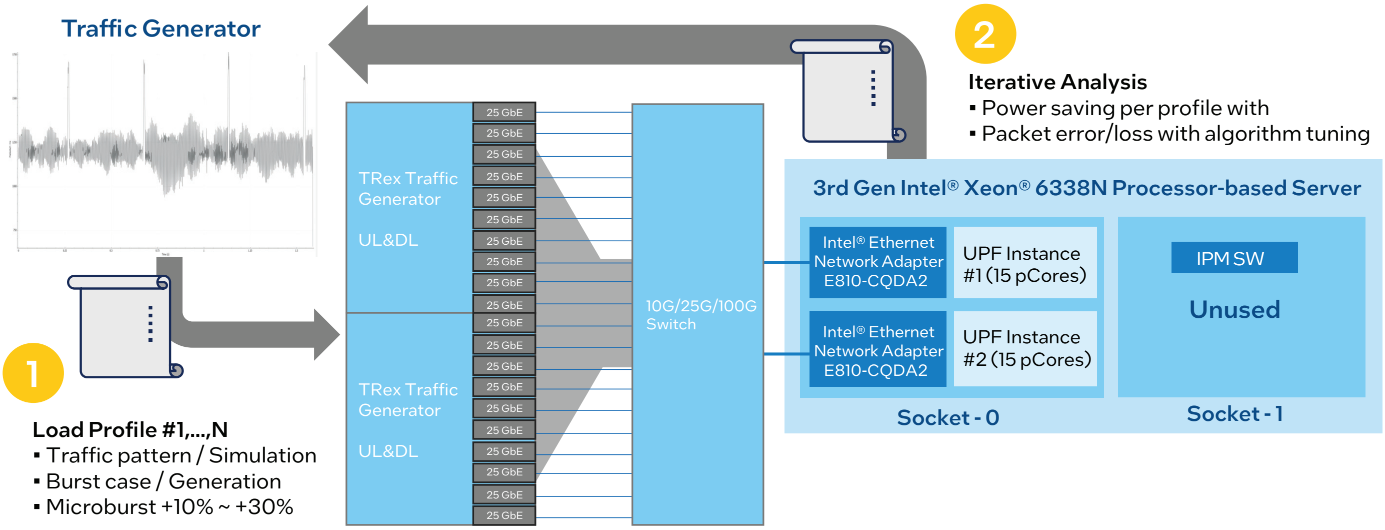
1. Packets entering the system from ports on the NIC are transferred by DMA into queues or are queued from a single thread pinned to a core, then consumed by some other thread or core in the packet processing network function.
2. The packet processing network function (e.g., UPF) uses APIs provided by the underlying packet processing software stack (DPDK, VPP) to retrieve packets in batches, and then processes them in sequence.
3. The cycles consumed between consecutive polls for packets are tracked as well as the number of packets returned in each poll to identify idle polls (i.e., no packets returned), and accordingly, the cycles consumed to execute real work versus cycles consumed to execute idle polls. This information is used to compute the real utilization on each core and is stored in an internal data structure in the context of the UPF.
4. The IPM periodically requests and obtains the CPU utilization status from the packet processing network function over either a socket interface (DPDK) or a shared memory interface (VPP).
5. The IPM implements an algorithm to compute the target frequency of each core while also ensuring there is sufficient hysteresis to prevent unnecessary oscillations. The CPU core frequency can be increased or decreased in steps of 100 MHz based on CPU utilization calculated over a sliding time window. To prevent packet errors or loss, our developed algorithm is tuned to favor CPU frequency increases over decreases.
6. Core frequency changes are done by enabling the Intel pstate driver in BIOS and providing hints to set the frequency at each lcore level using the Linux /sys/ filesystem.

The IPM also implements resiliency capabilities to ensure the high availability required in telco-grade applications, including the following:

- The IPM is initialized with peak performance settings and retains those settings until telemetry is successfully read and new target frequencies are computed.
- The IPM maintains peak performance while the IPM is gracefully terminated.
- The IPM ensures performance under burst conditions (0-100 Gbps instantaneous ramp with zero packet loss).
- The IPM maintains peak performance under conditions when telemetry data is temporarily unavailable.
- If the IPM fails or crashes, it is automatically restarted with configurable restart policies and timeout (e.g., 3 seconds).

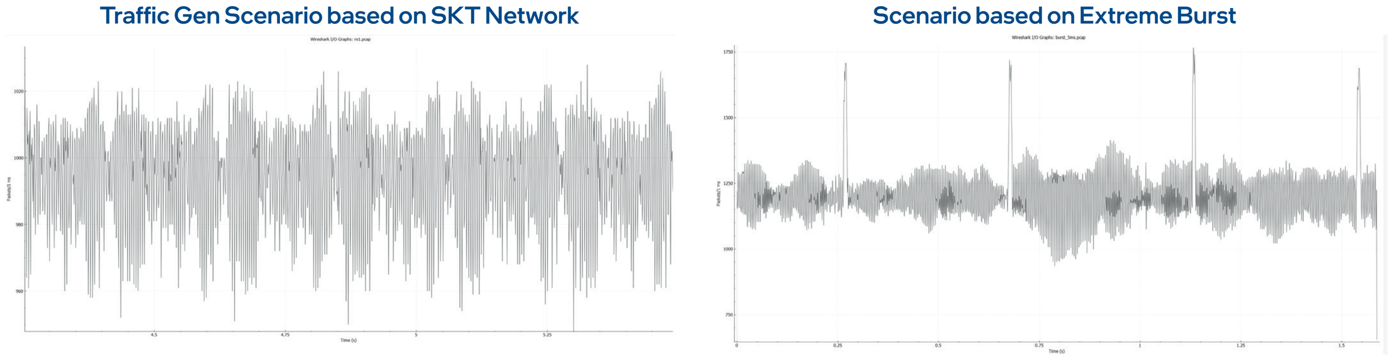
4.2 Testbed to exercise, validate and measure user plane power savings

We developed a test system to simulate high packet processing rates. This system programmatically generated packets that varied in rate, sizes and burst characteristics during a given time window. Figure 12 illustrates the setup for exercising the IPM. The packet processing network function used for these tests is the Intel FlexCore 5G UPF that implements 3GPP compliant functionalities (among other capabilities).



Traffic Generator
1
Load Profile #1,...,N
▪ Traffic pattern / Simulation
▪ Burst case / Generation
▪ Microburst +10% ~ +30%
TRex Traffic Generator
UL&DL
TRex Traffic Generator
UL&DL
25 GbE
10G/25G/100G Switch
2
Iterative Analysis
▪ Power saving per profile with
▪ Packet error/loss with algorithm tuning
3rd Gen Intel® Xeon® 6338N Processor-based Server
Intel® Ethernet Network Adapter E810-CQDA2
UPF Instance #1 (15 pCores)
Intel® Ethernet Network Adapter E810-CQDA2
UPF Instance #2 (15 pCores)
IPM SW
Unused
Socket - 0
Socket - 1

Figure 12. Test infrastructure for 5G User Plane power management


Traffic Gen Scenario based on SKT Network

Scenario based on Extreme Burst

Figure 13. Bursty nature of traffic in mobile networks, and burst traffic exercised in lab

The FlexCore 5G UPF ran two instances on a single CPU socket of a dual socket server, while the IPM ran on one core on another CPU socket. Two servers, each with 10 ports of 25GbE, generated simulated uplink and downlink traffic that was processed by the 5G UPF instances. Each UPF instance received uplink/downlink traffic from a dedicated software traffic generator (TRex) connected through an Ethernet switch. The characteristics of the traffic profile and the CPU frequencies in the test are as follows:

- PKT Size: 650 bytes, 3GPP UPF PDR/FAR/QER/URR processing using Intel FlexCore 5G User Plane Function (UPF).
- 50K UEs, 20 flows per UE (10 UL, 10 DL); total of 1M flows.
- 1:3 UL to DL packet rate/TPT ratio.
- Update interval (reads of CPU utilization from UPF) = 10 msec.
- Tx/Rx descriptor size = 2048, queue utilization included in UPF under test.
- CPU: 3rd Generation Intel Xeon Processor 6338N (32 cores, 2.2 GHz P1 frequency).
- CPU core frequency (min) = 800 MHz (configured by IPM software).
- CPU core frequency (max) = 2200 MHz (3rd Gen Intel® Xeon® 6338N CPU rated P1 frequency).
- P-states supported: 15 (800 MHz to 2200 MHz in 100 MHz steps).
- Memory: DDR4-2667 MHz (fixed).

Traffic in real mobile networks tends to be bursty in nature. As such, the traffic generator used in our tests generated bursty traffic as shown in Figure 13. The left side of the diagram illustrates default bursty traffic, while the one on the right side of the diagram illustrates packet capture of real, explicitly induced extreme bursts to stress test the system's ability to process packets with zero packet loss as the key performance indicator while power management is being applied simultaneously.

Burst conditions were applied to the tests in periods of about 5-millisecond duration above the nominal packet rate. Each burst could increase the packet rate up to 30% over the nominal rate for up to 50% of the peak load, and up to 20% increase over the nominal rate for up to 75% of peak load. Figures 14-16 illustrate a 24-hour load profile with different max loading and burst considerations.

Based on the profiles above, TRex traffic scripts were created to emulate real network traffic conditions. Each hour was compressed into one minute, requiring 24 minutes for one complete test run. The Intel Power Thermal Utility (Intel PTU) was used to capture the current CPU frequency and CPU power consumption at the socket level at one-second intervals. Packet drop statistics were also collected after each minute from the two UPF instances.

4.3 Power Measurements and Results

Tests were conducted for five different scenarios to measure power savings and packet losses:

- Traffic Profile #1: 30% max loading on the UPF (bursts included in each of the 24-hr intervals).
- Traffic Profile #2: 50% max loading on the UPF (bursts included in each of the 24-hr intervals).
- Traffic Profile #3: 75% max loading on the UPF (bursts included in each of the 24-hr intervals).
- Impact of number of bursts (1, 5, 60 bursts).
- Validating worst case, random burst of 0 packets per second (0 Gbps) ramped to 18 million packets per second (100 Gbps) in under 1 second.

4.3.1 Test Results for Traffic Profiles

The graphs shown in Figures 14, 15, and 16 show the test results for the three traffic profiles above. These were measured with zero packet loss.

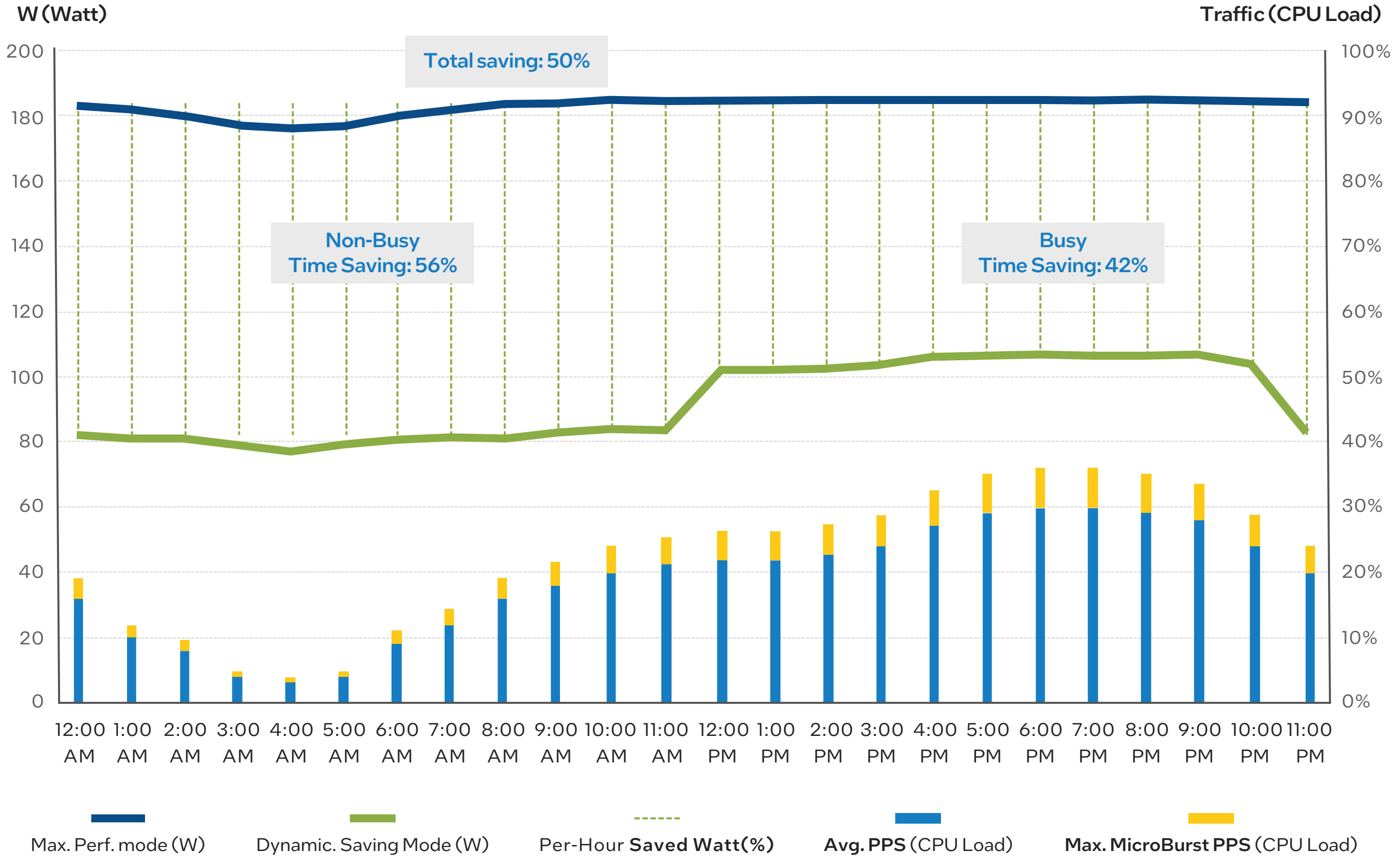


Figure 14. Result for Load Profile#1 (CPU Util. ~30%)

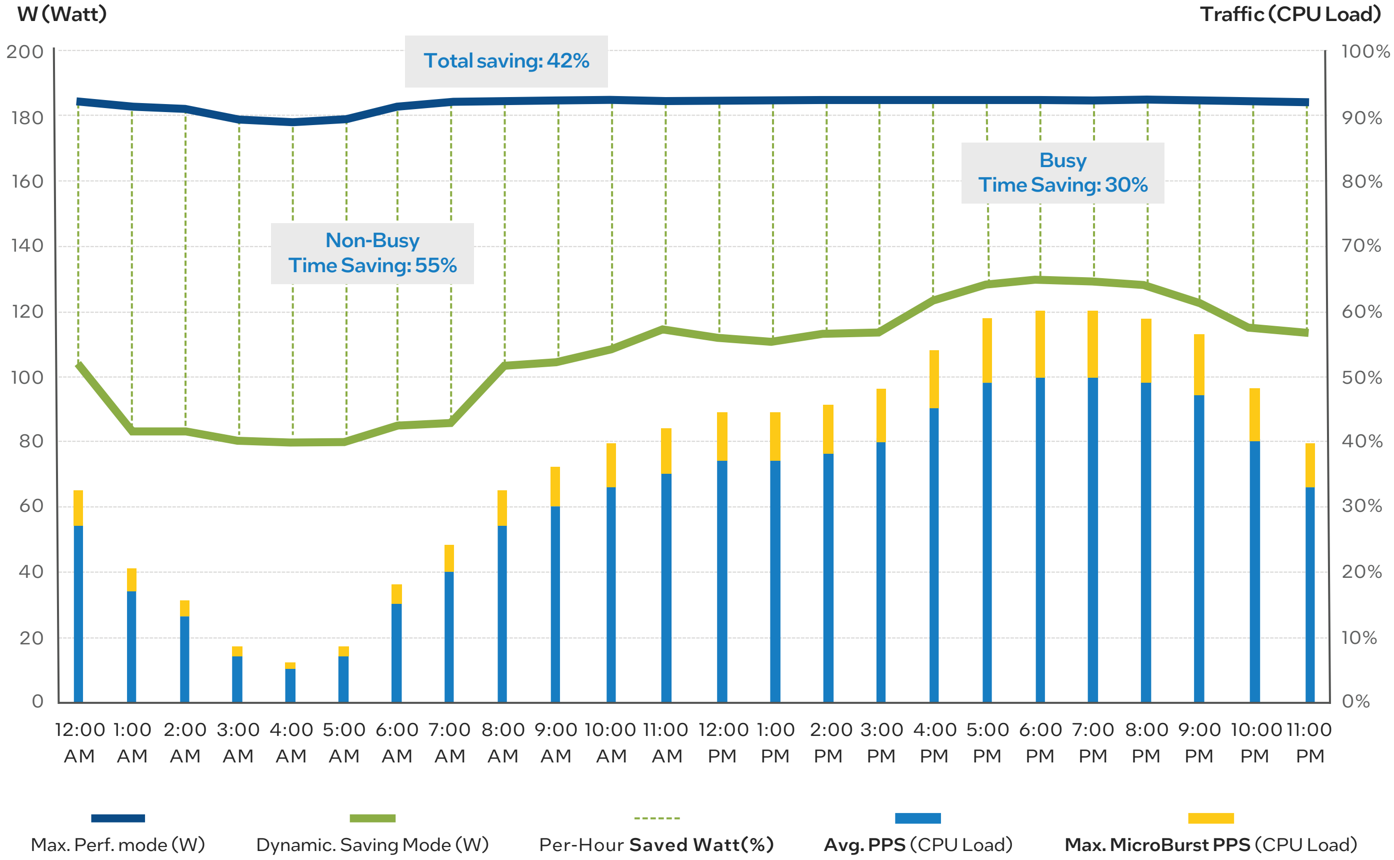


Figure 15. Result for Load Profile#2 (CPU Util. ~50%)

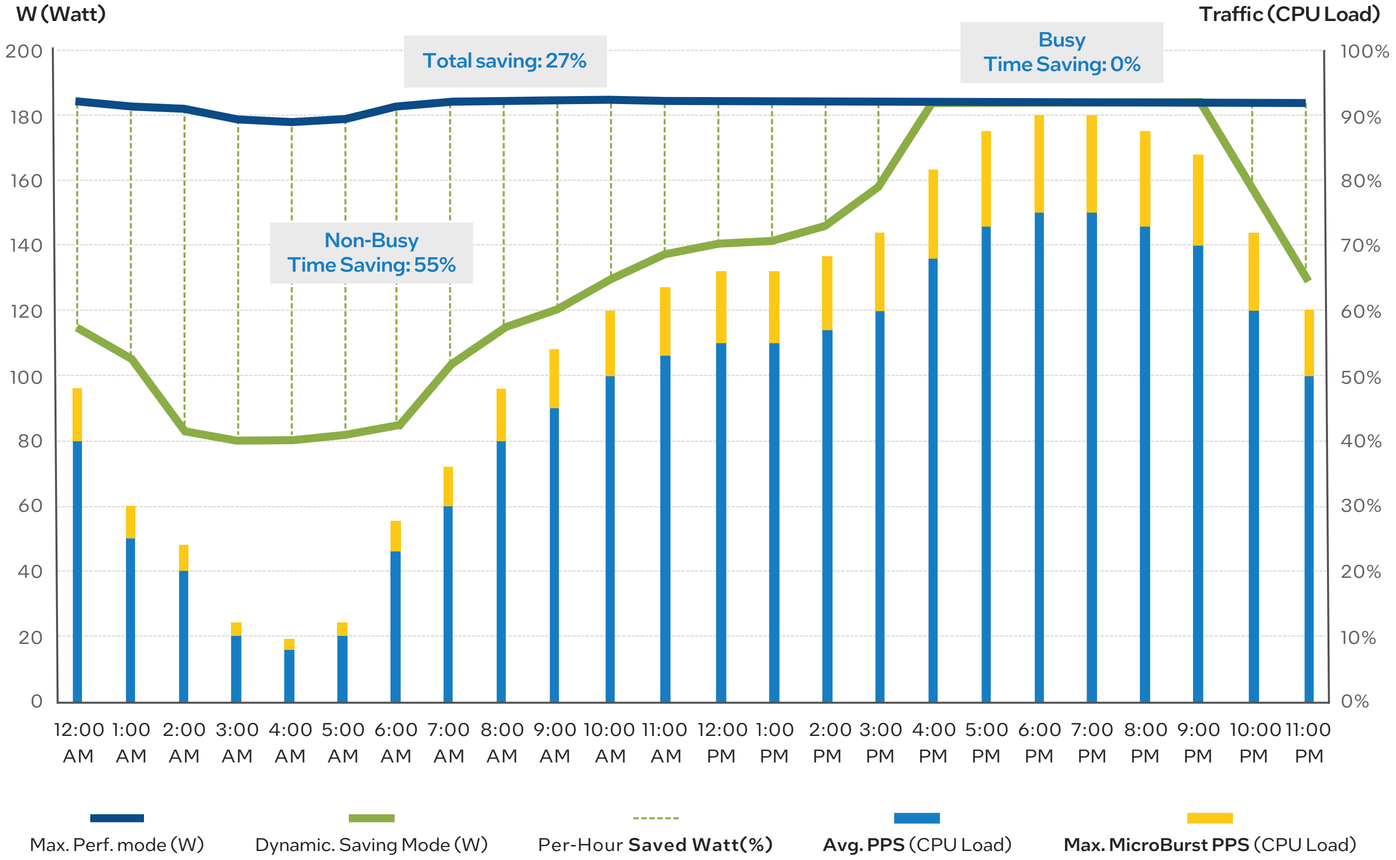


Figure 16. Result for Load Profile #3 (CPU Utilization ~75%)

Table 4. Power savings for different traffic profiles

	Non-busy hour savings	Busy hour savings	Average savings over 24-hour period
Baseline (Without IPM)	None (~180W CPU power consumption)	None (~185W CPU power consumption)	None
Traffic Profile #1	56%	42%	50%
Traffic Profile #2	55%	30%	42%
Traffic Profile #3	55%	None	27%

4.3.2 Impact of Power Savings by Varying Number of Bursts

One of the key analyses conducted was to evaluate the sensitivity of number of bursts to overall power savings, as burst conditions require inherent increase in CPU core frequencies to process packets successfully without incurring packet losses. Towards that, traffic profile #3 was instrumented with different number of bursts and its impact to power savings were measured. The number of bursts per interval was changed between 1, 30, or 60 bursts per 1-min interval. These tests are indicative of ensuring power savings even with varying number of bursts, wherein only a 2% increase in average power consumption was measured even with worst case of 60 bursts per one-minute intervals, thereby ensuring power savings across all traffic conditions.

4.3.3 Worst Case, Random Burst of 100 Gbps/18 MPPS

To simulate worst case traffic conditions, a synthetic, lab-oriented test was constructed wherein the UPF executed in an idle condition (i.e., 0 PPS), with the IPM managing its power. Under this idle condition, the IPM would be expected to lower the core frequency of the packet processing cores to Pn (i.e., 800 MHz for the CPU under test). While in this condition, the traffic generator randomly initiated a traffic ramp-up within one second to 18 million packets per second. This ramp-up of traffic was completed within one second, as shown in Figure 17.

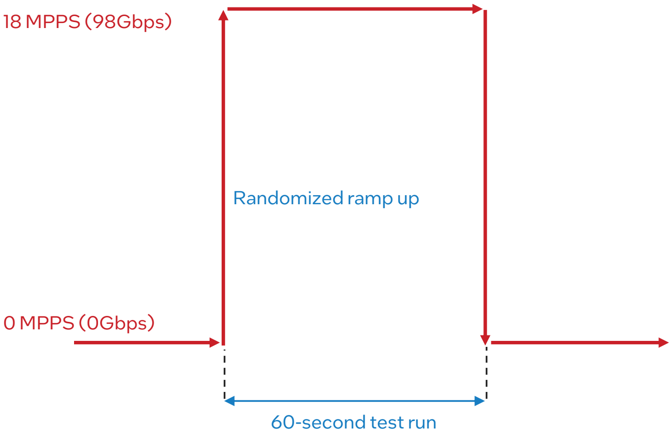


Figure 17. Worst case traffic burst test condition

We observed that the UPF was able to buffer sufficient packets into the system (by tuning the receive queue length to 2048 per queue), and that the IPM was able to track the increase of utilization on the cores. The IPM successfully increased the frequency from Pn (800 MHz) to P1 (2200 MHz) such that no packet drops occurred in the system. This demonstrates that the overall architecture and implementation can handle any unforeseen traffic conditions not already exercised as part of this work.

4.3.4 Server hardware Configuration used for tests

Table 5. Server hardware configuration for user plane power management tests

Ingredient	Description
Processor	Two 3rd Generation Intel® Xeon® 6338N Processors (32 cores at 2.2 GHz)
Memory	128 Gbytes per CPU, 256 Gbytes total
Network I/O	Intel® Ethernet Network Adapter E810-CQDA2 (qty=2)

5. Methodology for Dynamic Power Saving in Control Plane Network Functions

5.1 Control/Management Plane Function (CPF)

CPF workloads typically process signaling transactions which may not have very high packet rates but are nevertheless sensitive to processing latencies, and more likely than not, use a kernel networking stack for communication within or across applications. Examples of such workloads include Access and Mobility Management Function (AMF), Session Management Function (SMF), Policy Control Function (PCF), Charging Function (CHF), Unified Data Management (UDM) and Network Exposure Function (NEF), as shown in Figure 18.

These categories of workloads execute as regular applications in Linux user space with a Linux kernel networking stack. Such workloads also oversubscribe cores wherein the Linux scheduler makes runtime decisions to schedule threads onto cores, as well as preempting threads to schedule other threads with one of the various scheduling algorithms available in the Linux scheduler. The kernel has good observability into each core's utilization, which it uses to make scheduling decisions (at 10-msec granularity by default). The kernel also has hooks into processor-architecture-dependent power management drivers to manage C-states, P-states or both.

5.2 Implementation Approach

Control plane network functions are designed and deployed as microservices that may include a service mesh such as Istio/Envoy. A typical 5G core contains a broad range of network functions, each further decomposed into microservices. The breadth of microservices can vary significantly, and instances of microservices can dynamically change over time to account for scaling functionalities to handle ever changing loads in the network.

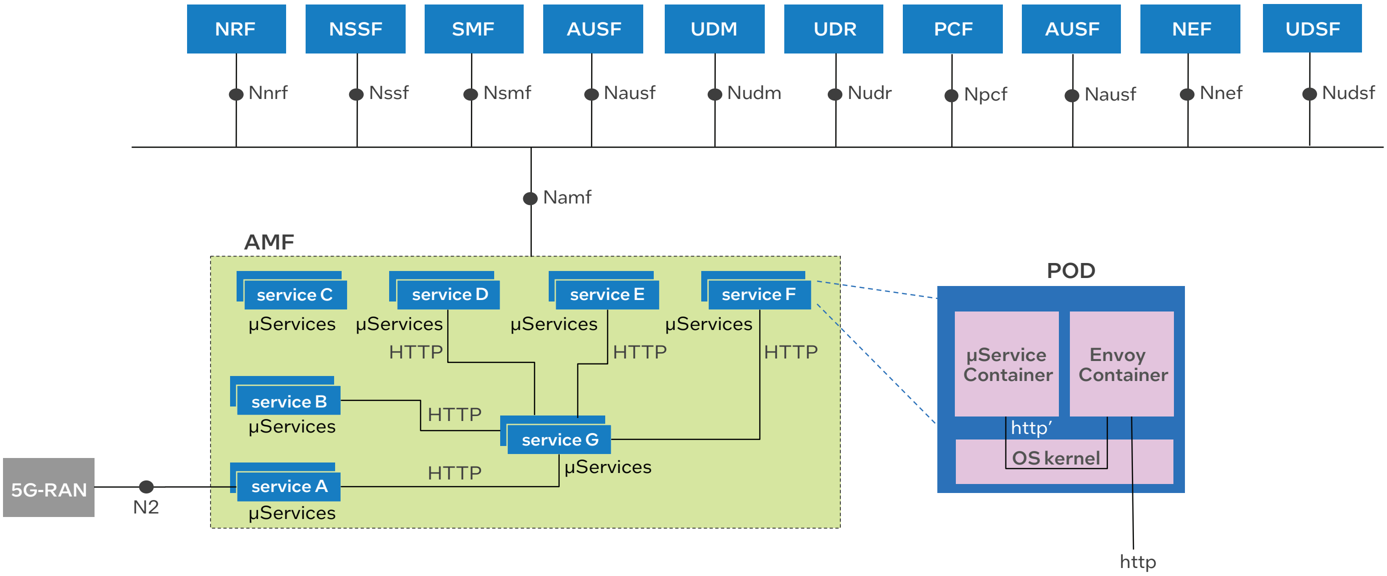


Figure 18. 5G Control Plane Network Functions implemented with Cloud Native design principles

As such, the approach taken for control plane power management (CPM) is independent of the 5G core control plane network functions based on microservices and is a user space application that runs on all nodes of a Kubernetes cluster that have 5G core network functions deployed on them.

CPM evaluates CPU utilization at runtime and determines the optimal uncore frequency configuration. Complementary to CPM, the CPU frequency scaling governor is also enabled in the operating system to save power by changing the CPU P-states, and the C-states via the ACPI driver. These two capabilities form the basis for control plane power management and remove the need for costly OS-based CPU frequency scaling governor changes in the kernel.

5.3 Test Bed and Scenarios

In order to simulate realistic 5G control plane transactions, two control plane transactions were considered: UE registration and deregistration request, and PDU session establishment and release. Call flows associated with these transactions are illustrated in Figures 19 through 22.

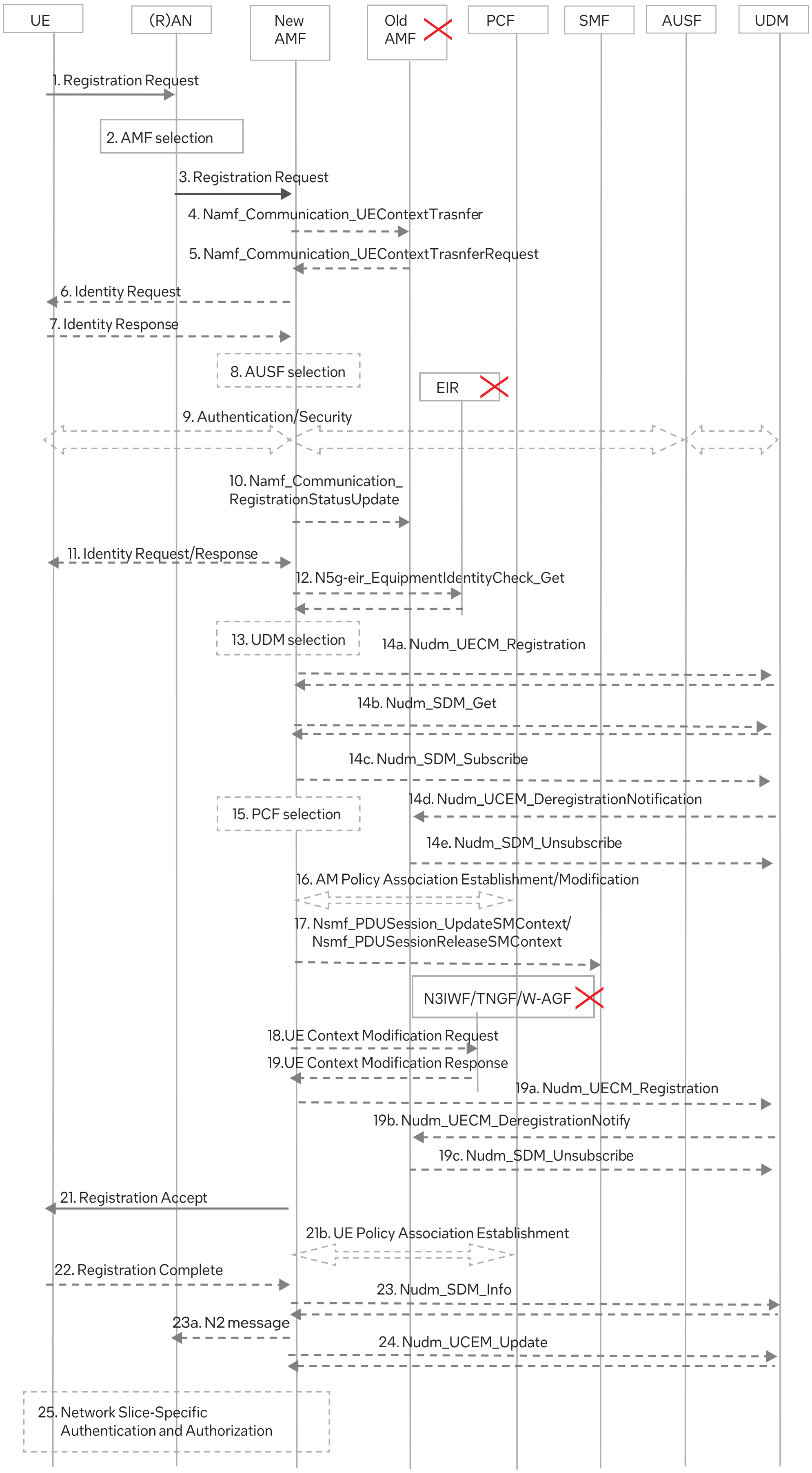

UE
(R)AN
New AMF
Old AMF
PCF
SMF
AUSF
UDM
1. Registration Request
2. AMF selection
3. Registration Request
4. Namf_Communication_UEContextTrasnfer
5. Namf_Communication_UEContextTrasnferRequest
6. Identity Request
7. Identity Response
8. AUSF selection
EIR
9. Authentication/Security
10. Namf_Communication_RegistrationStatusUpdate
11. Identity Request/Response
12. N5g-eir_EquipmentIdentityCheck_Get
13. UDM selection
14a. Nudm_UECM_Registration
14b. Nudm_SDM_Get
14c. Nudm_SDM_Subscribe
14d. Nudm_UCEM_DeregistrationNotification
15. PCF selection
14e. Nudm_SDM_Unsubscribe
16. AM Policy Association Establishment/Modification
17. Nsmf_PDUSession_UpdateSMContext/ Nsmf_PDUSessionReleaseSMContext
N3IWF/TNGF/W-AGF
18.UE Context Modification Request
19.UE Context Modification Response
19a. Nudm_UECM_Registration
19b. Nudm_UECM_DeregistrationNotify
19c. Nudm_SDM_Unsubscribe
21. Registration Accept
21b. UE Policy Association Establishment
22. Registration Complete
23. Nudm_SDM_Info
23a. N2 message
24. Nudm_UCEM_Update
25. Network Slice-Specific Authentication and Authorization

Figure 19. UE registration call flow

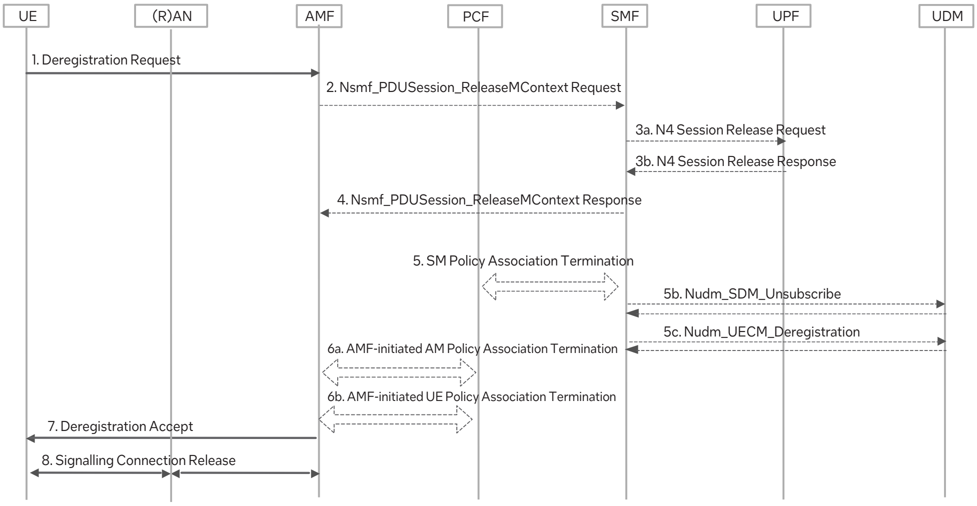

UE
(R)AN
AMF
PCF
SMF
UPF
UDM
1. Deregistration Request
2. Nsmf_PDUSession_ReleaseMContext Request
3a. N4 Session Release Request
3b. N4 Session Release Response
4. Nsmf_PDUSession_ReleaseMContext Response
5. SM Policy Association Termination
5b. Nudm_SDM_Unsubscribe
5c. Nudm_UECM_Deregistration
6a. AMF-initiated AM Policy Association Termination
6b. AMF-initiated UE Policy Association Termination
7. Deregistration Accept
8. Signalling Connection Release

Figure 20. UE deregistration call flow

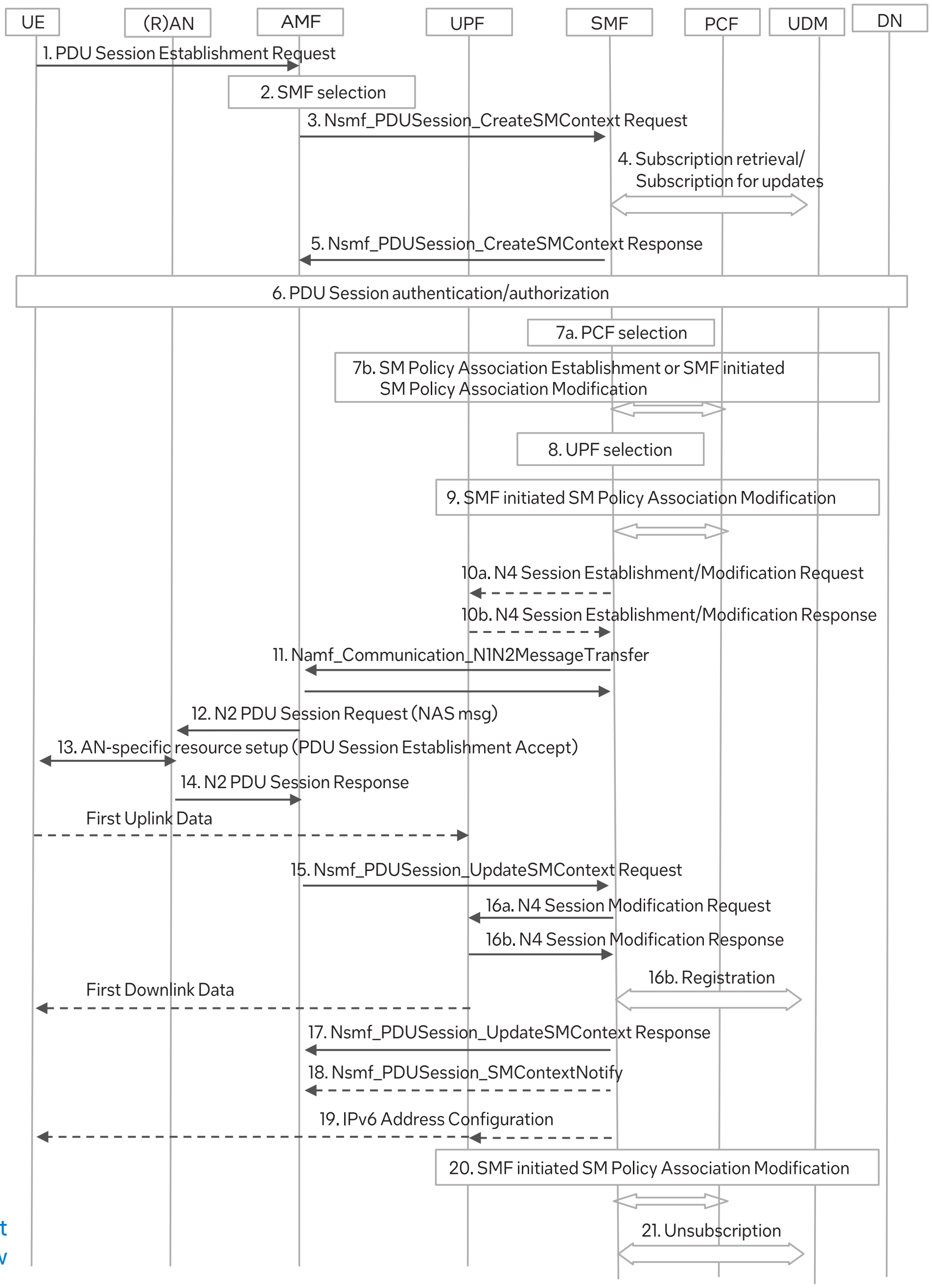

UE
(R)AN
AMF
UPF
SMF
PCF
UDM
DN
1. PDU Session Establishment Request
2. SMF selection
3. Nsmf_PDUSession_CreateSMContext Request
4. Subscription retrieval/
Subscription for updates
5. Nsmf_PDUSession_CreateSMContext Response
6. PDU Session authentication/authorization
7a. PCF selection
7b. SM Policy Association Establishment or SMF initiated
SM Policy Association Modification
8. UPF selection
9. SMF initiated SM Policy Association Modification
10a. N4 Session Establishment/Modification Request
10b. N4 Session Establishment/Modification Response
11. Namf_Communication_N1N2MessageTransfer
12. N2 PDU Session Request (NAS msg)
13. AN-specific resource setup (PDU Session Establishment Accept)
14. N2 PDU Session Response
First Uplink Data
15. Nsmf_PDUSession_UpdateSMContext Request
16a. N4 Session Modification Request
16b. N4 Session Modification Response
16b. Registration
First Downlink Data
17. Nsmf_PDUSession_UpdateSMContext Response
18. Nsmf_PDUSession_SMContextNotify
19. IPv6 Address Configuration
20. SMF initiated SM Policy Association Modification
21. Unsubscription

Figure 21. UE session establishment call flow

These call flows were exercised in a test as illustrated in Figure 23, comprising a Kubernetes cluster of two dual socket servers based on Intel Xeon 6338 CPUs. On this cluster, a 5G core software stack based on cloud-native microservices was deployed as bare metal containers. It was essentially a system under test (SUT) on an off-the-shelf operating system (CentOS) with CaaS and PaaS ingredients.

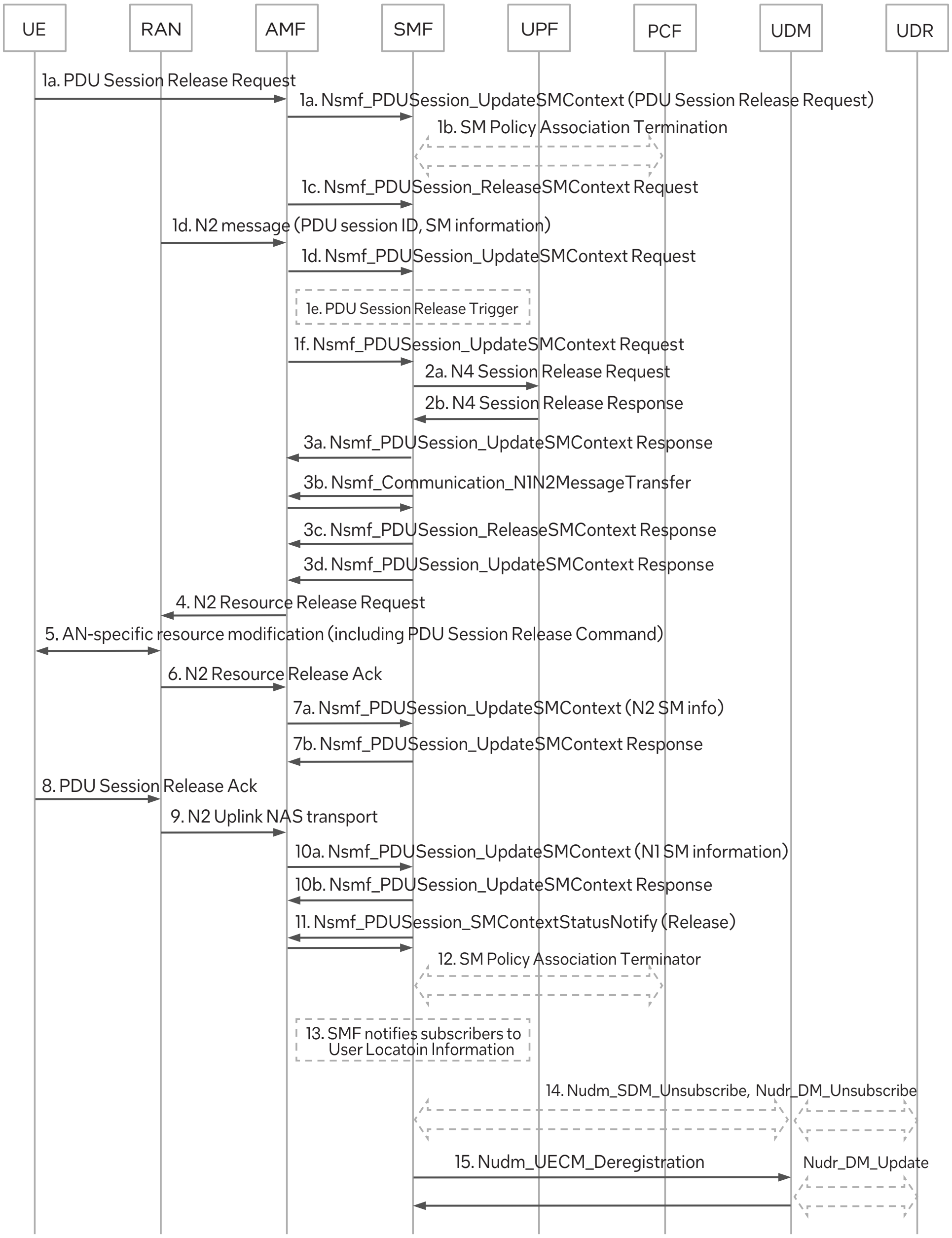

UE
RAN
AMF
SMF
UPF
PCF
UDM
UDR
1a. PDU Session Release Request
1a. Nsmf_PDUSession_UpdateSMContext (PDU Session Release Request)
1b. SM Policy Association Termination
1c. Nsmf_PDUSession_ReleaseSMContext Request
1d. N2 message (PDU session ID, SM information)
1d. Nsmf_PDUSession_UpdateSMContext Request
1e. PDU Session Release Trigger
1f. Nsmf_PDUSession_UpdateSMContext Request
2a. N4 Session Release Request
2b. N4 Session Release Response
3a. Nsmf_PDUSession_UpdateSMContext Response
3b. Nsmf_Communication_N1N2MessageTransfer
3c. Nsmf_PDUSession_ReleaseSMContext Response
3d. Nsmf_PDUSession_UpdateSMContext Response
4. N2 Resource Release Request
5. AN-specific resource modification (including PDU Session Release Command)
6. N2 Resource Release Ack
7a. Nsmf_PDUSession_UpdateSMContext (N2 SM info)
7b. Nsmf_PDUSession_UpdateSMContext Response
8. PDU Session Release Ack
9. N2 Uplink NAS transport
10a. Nsmf_PDUSession_UpdateSMContext (N1 SM information)
10b. Nsmf_PDUSession_UpdateSMContext Response
11. Nsmf_PDUSession_SMContextStatusNotify (Release)
12. SM Policy Association Terminator
13. SMF notifies subscribers to User Locatoin Information
14. Nudm_SDM_Unsubscribe, Nudr_DM_Unsubscribe
15. Nudm_UECM_Deregistration
Nudr_DM_Update

Figure 22. UE session release call flow

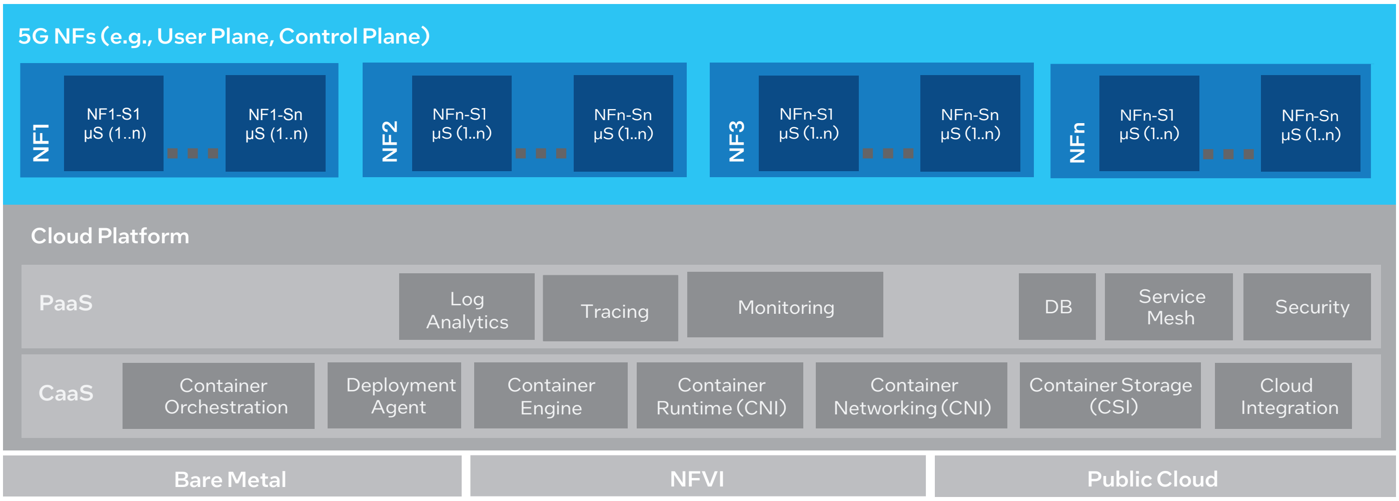


Figure 23. Cloud Native Platform Infrastructure for 5G Core

Table 6 shows the 5G core containerized network functions deployed in the two-node Kubernetes cluster, exercised for power saving measurements:

Table 6. 5GC Network Functions covered in test infrastructure for control plane power management

No.	NF	Description / Comments
1.	Access and Mobility Management Function (AMF)	NAS termination, NAS ciphering/integrity protection, connection, mobility management
2.	Session Management Function (SMF)	Session mgmt. (establishment/modification/release), IP address allocation, configuring UPF for routing/steering, etc.
3.	Authentication Server Function (AUSF)	Provides authentication server capability, interacts with AMF, UDM, etc.
4.	Policy Management Function (PCF)	Provides policy rules such as access subscription information for policy decisions in UDR
5.	Unified Data Management (UDM)	Generates authentication and key agreement (AKA) credentials, user auth., access authorization, subscription mgmt.
6.	Unified Data Repository (UDR)	Centralized storage of subscription information used by other NFs
7.	Unstructured Data Storage Function (UDSF)	Stores unstructured data, session states, dynamic information, etc.
8.	Network Slice Selection Function (NSSF)	Selection of network slice instances to UE, allowed NSSAI, etc.
9.	Network Repository Function (NRF)	Service registration, discovery, maintains NF profile and instances
10.	Network Exposure Function (NEF)	Exposure of capabilities, events
11.	User Plane Function (UPF)	Packet routing, forwarding, inspection, policy enforcement, session anchor, etc. UPF will include NW-TT for TSN

Table 7 shows the CaaS and PaaS ingredients on the test platform that are integrated with the 5G CNFs:

Table 7. CaaS/PaaS platform ingredients

No.	NF	Function	Description / Comments
1.	CaaS ingredients	Container orchestration	Kubernetes
2.			kubernetes dashboard
3.		Container engine	Docker
4.		Container runtime	Containerd
5.		Container networking	Calico
6.			Multus
7.			SR-IOV CNI
8.			SR-IOV Device Plugin
9.			Luigi
10.			Macvlan
11.			Whereabouts
12.			coredns
13.	PaaS ingredients	Container storage	Commercial product
14.		Package management	Helm
15.		Service mesh - Istio	istio/operator
16.			istio/pilot
17.			istio/proxyv2
18.		Kiali	kiali
19.		Image and artifact repository	Harbor
20.		Logging – fluentd	fluent-bit
21.		Dashboard	Kibana
22.		Elastic search	eck-operator
23.			es-curator
24.			elasticsearch_exporter
25.			metricbeat
26.			alertmanager
27.			elasticsearch
28.		Jaeger	jaegertracing/all-in-one
29.		K8s sidecar	k8s-sidecar
30.		Prometheus	kube-state-metrics
31.			node-exporter
32.			prometheus-config-reloader
33.			prometheus-operator
34.			prometheus
35.			alertmanager
36.			kube-webhook-certgen
37.			configmap-reload
38.			grafana
39.		Traefik	traefik
40.		ETCD	etcd-alpine
41.		Zookeeper	zk-alpine
42.		Kafka	Kafka-apline

These CaaS/PaaS platform infrastructure ingredients and the containerized microservices-based 5G network functions were deployed on a two-node Kubernetes cluster based on dual socket 3rd Generation Intel Xeon 6338 processors. These processors have 32 cores, with a rated core frequency of 2 GHz and a maximum thermal design point (TDP) of 205W. These two nodes have a total of four processors on which the Kubernetes infrastructure schedules microservice pods as shown in Figure 24. These nodes are interconnected by a common 10G/25G/100G Ethernet switch, to which a 5G RAN/UE traffic emulator is also connected. The RAN/UE traffic emulator used is Spirent Landslide, which can be programmed to emulate a varying number of gNBs, UEs and its associated control plane signaling call flows described in Figures 19, 20, 21, and 22.

To be able to characterize the potential power savings, three realistic workload scenarios were considered representing load profiles across a 24-hour period. These load profiles were based on actual loads in a mobile network, representing transactions generated by user equipment (UEs) into the 5G core. The load profiles used for measuring power savings are shown in Figure 25.

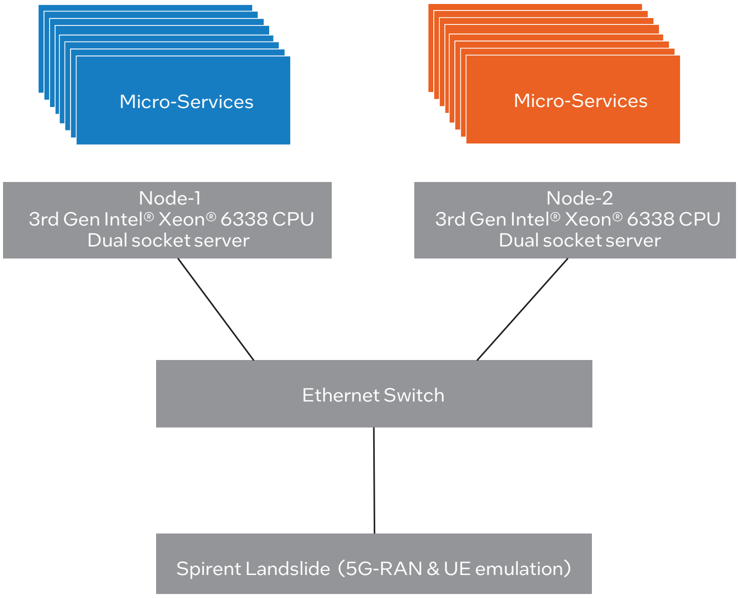

Micro-Services
Micro-Services
Node-1
3rd Gen Intel® Xeon® 6338 CPU
Dual socket server
Node-2
3rd Gen Intel® Xeon® 6338 CPU
Dual socket server
Ethernet Switch
Spirent Landslide (5G-RAN & UE emulation)

Figure 24. Test setup structure for 5G Control Plane power management

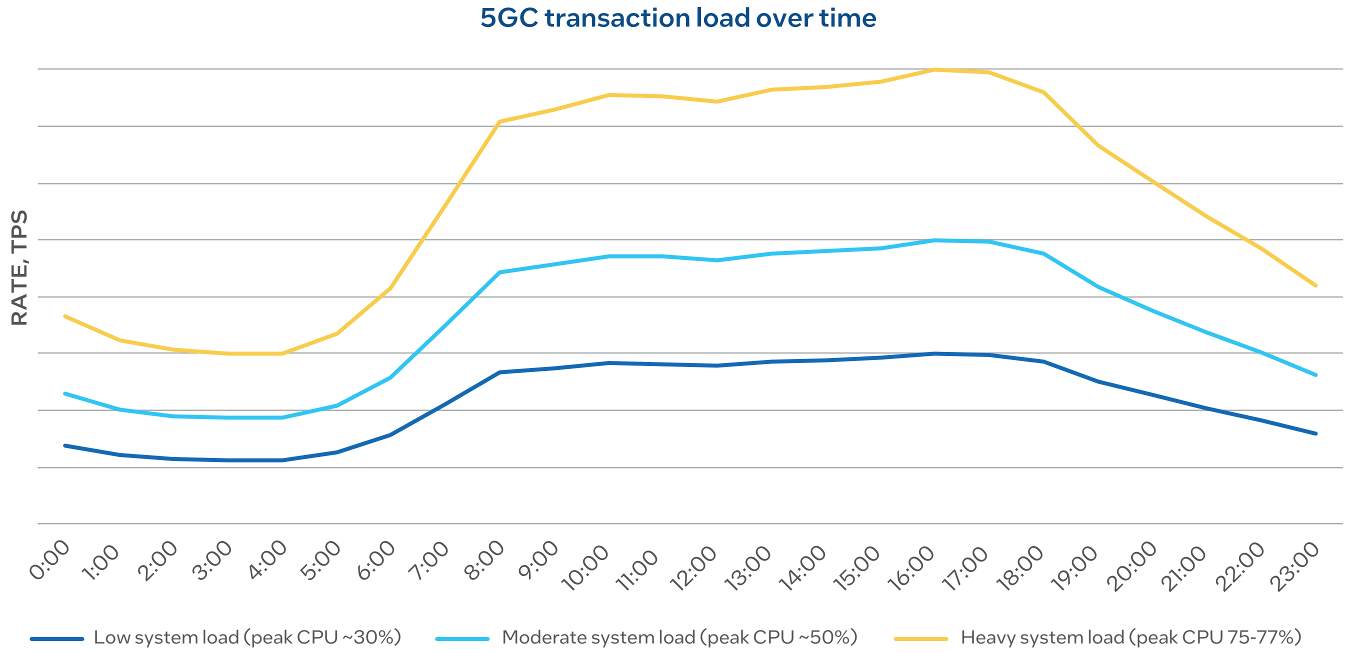


Figure 25. 5G Control Plane traffic profiles exercised for power savings measurements

5.4 Power Measurements and Results

The 24-hour traffic profiles described in Figure 25 were exercised with and without control plane power management. The associated CPU core frequencies, uncore frequencies and power consumption were recorded to evaluate the overall maximum performance.

Table 8. Configurations for control plane power savings measurements

	Core C1, C6 states	P-states	Uncore frequency
Baseline (default)	Disabled	Disabled	Fixed
With CPM	Enabled	Enabled	Variable

Historically, NFVi systems have been deployed with the baseline configuration shown in Table 8 to realize maximum performance in the control plane. With CPM, C1, C6 and P-states are available, along with power savings based on uncore frequency changes.

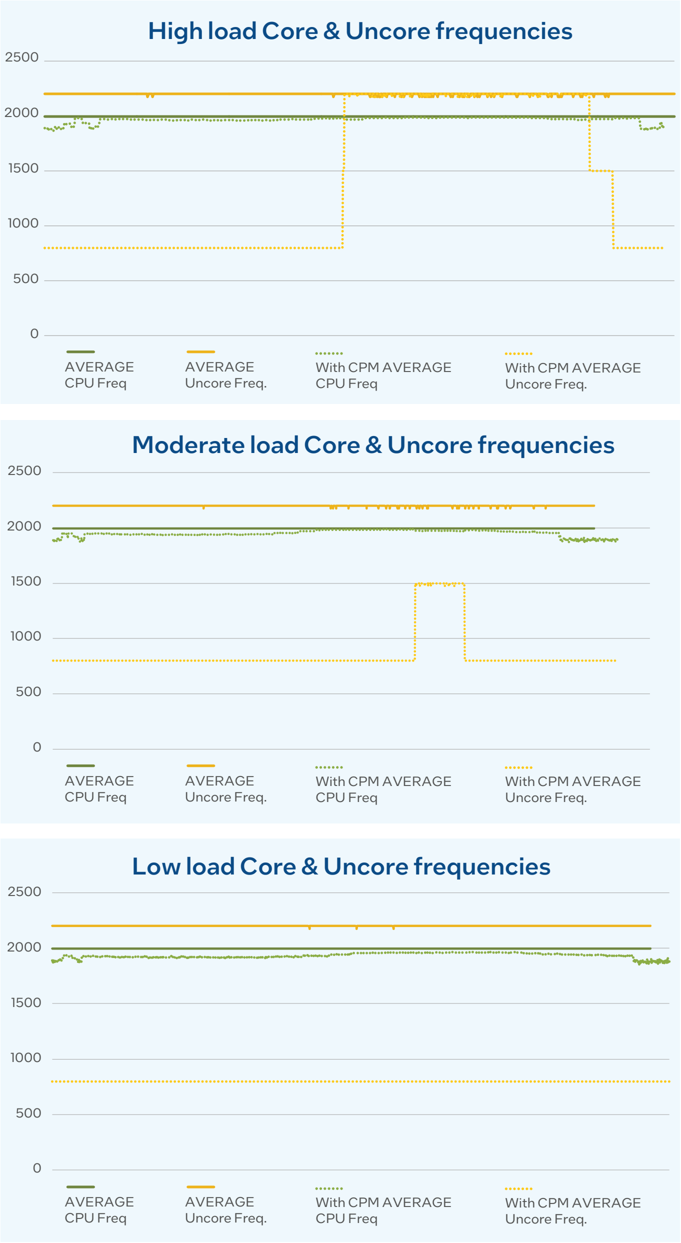


Figure 26. Core and Uncore frequencies across different traffic profiles with control plane power management

5.4.1 CPU Frequencies and C/P-state Residencies

Figure 26 shows the CPU frequency and the C- and P-state under high, moderate and low-load conditions. For the duration of the tests, the uncore frequency was automatically adjusted in response to the transactions in the system. During busy hour conditions, we observed that the uncore frequencies of the CPUs were increased, while the CPU core frequencies were maintained at the rated frequencies.

For the high load condition, the uncore frequency was increased to 2200 MHz for a longer duration. For the moderate load condition, the duration of frequency increases was reduced. Under a low load scenario, we observed that the uncore frequency stayed at the minimum for duration of the test. This forms the primary power saving capability for these scenarios, enabled by the control plane power manager prototype implementation.

The secondary source of power saving is derived from the C1/C6 power states driven by the operating system. We evaluated a range of operating system power saving governors for overall power saving and KPI impact. To enable the lowest possible KPI impact and to maximize power savings, the performance power governor was selected, allowing the operating system to transition cores into C1- or C6-state. The CPU C0/C1 and C6 residencies for the three scenarios exercised are shown in Figure 27. These charts show an average residency of these states across all cores of all four CPUs in the test.

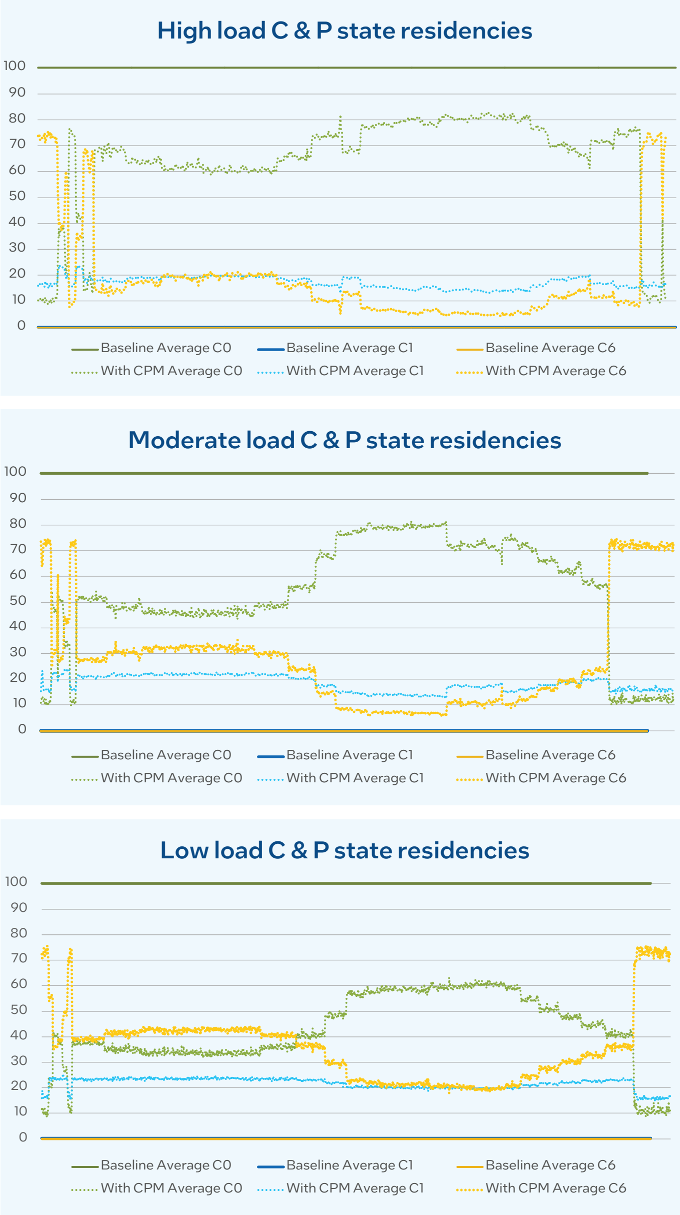


Figure 27. C0, C1, and C6 residencies for high, moderate and low load scenarios

5.4.2 Power Savings

For 5G control plane workloads, control plane power management (CPM) resulted in a runtime average power savings from about 15% under a high load profile to about 40% under a low load profile, as seen in Figure 28.

The plots in Figure 29 show the power savings for the duration of the test for each of the traffic profiles. The X axis indicates hourly intervals in a representative 24-hour period during which the load changed based on the time of day. The primary vertical axis shows the power consumption as an average value across four CPUs. The secondary vertical axis shows the control plane load as a percentage. The black lines indicate the baseline configuration power consumption. The green lines indicate the runtime power savings enabled by control plane power management. The orange lines indicate the power savings possible using only the operating system's CPU frequency scaling governor. When power saving mode was turned on, control plane per TPS's show increased latencies, though no failures were observed, as determined by timing parameters defined in 3GPP specifications. Depending on latency sensitivity and or deployment scenarios, this may not be significant, as over the air latency typically is a dominant part in the total transaction latency between UE and the 5G core.

These results demonstrate that CPU power savings are possible in the range of up to 40% by leveraging capabilities of CPM combined with the CPU frequency scaling governor built into the Linux operating system.

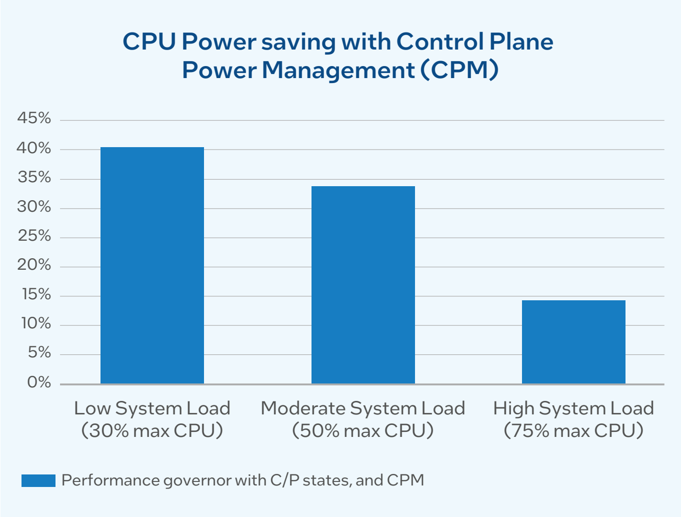


Figure 28. Average control plane power savings across load profiles

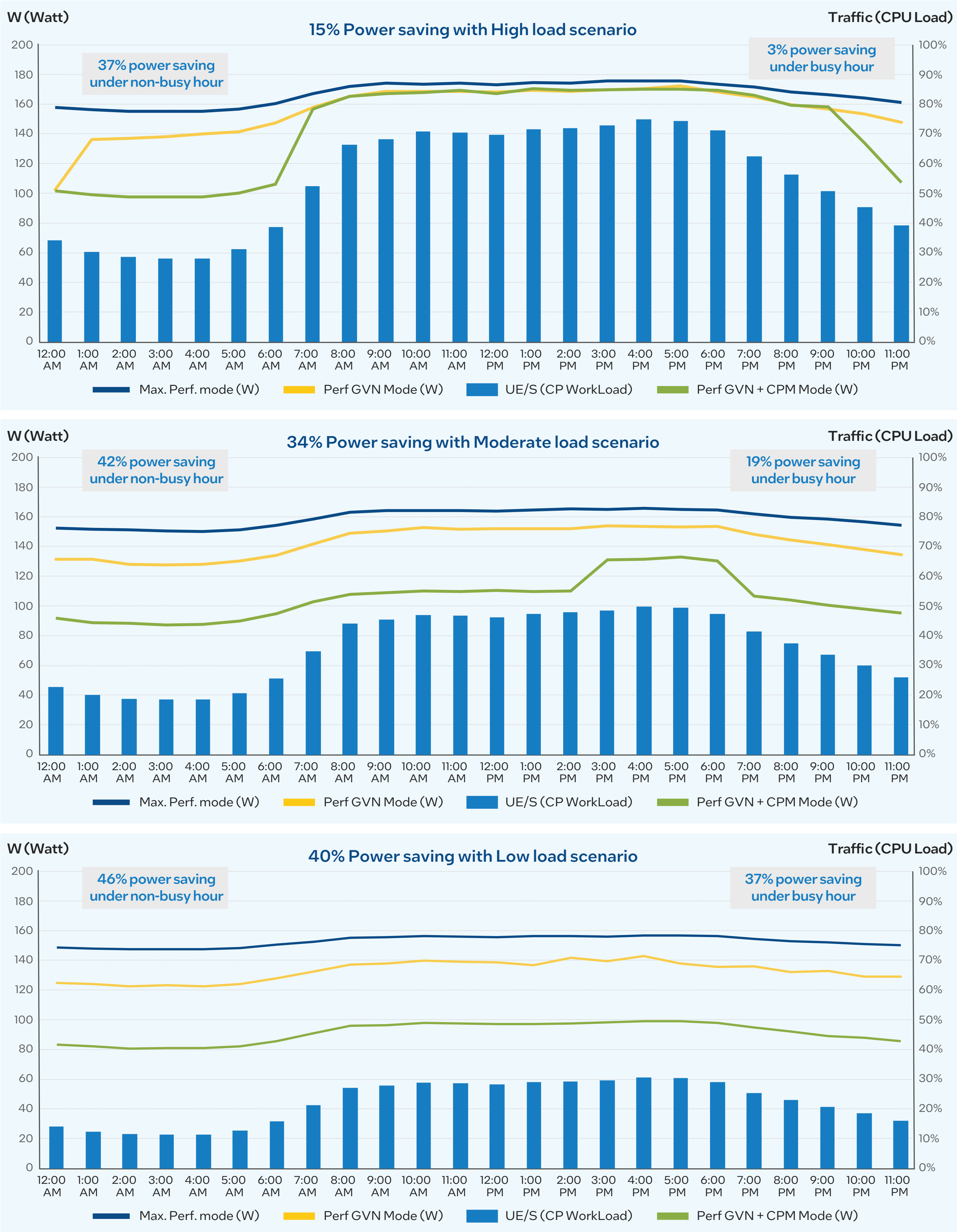


Figure 29. Control Plane power savings for high, moderate and low load scenarios

5.4.3 Server Hardware Configuration Used for Tests

Table 9. Hardware configuration for control plane power management tests

Ingredient	Description
Processor	Two 3rd Generation Intel® Xeon® 6338 Processor (32 cores at 2 GHz)
Memory	512 Gbytes total
Network I/O	Intel® Ethernet Network Adapter E810-CQDA2 (qty=2)

Summary

This paper demonstrates that dynamic power saving technology can save overall energy consumption in cloud-native 5G core infrastructure. It can optimize energy consumption without compromising key performance indicators such as throughput, latency and packet loss by utilizing the telemetry capabilities for constantly fluctuating commercial traffic in real time.

Using the dynamic power saving techniques in SK Telecom’s traffic model for user plane, we demonstrated that power consumption can be reduced up to 55% during the non-busy hours and 30% during the busy hours, resulting in a 42% reduction over a 24-hour period.

Likewise, the power consumption can be reduced up to 42% during the non-busy hours and 19% during the busy hours, resulting in a 34% reduction over the 24-hour period in SK Telecom's traffic model for control plane. These capabilities show it is possible for COSPs to utilize Intel dynamic energy saving technology to save energy consumption and reduce greenhouse gas emissions.

Making the commercial system to take advantage of this newly developed technique for a live benchmark study is important, and further research may be needed to support backward compatibilities to work with VNF-based systems. Looking forward, Intel and SK Telecom will continue to collaborate on the Beyond 5G and 6G Core research with software and hardware technology for energy efficiency and reducing overall greenhouse gas emissions.

Performance varies by use, configuration and other factors. Learn more at https://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 configuration disclosure for configuration details. No product or component can be absolutely secure.

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

Your costs and results may vary.

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

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 nonexclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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.

0223/BB/MESH/PDF 354467-001US