Computer Networks 270 (2025) 111487

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A Management Data Analytics Function for ethical 6G networks

Milad Akbari [a], Raffaele Bolla [a,b], Roberto Bruschi [a,b], Chiara Lombardo [a,b], Nicole Simone Martinelli [b], Beatrice Siccardi [a],*

[a] DITEN, University of Genoa, Genoa, Italy
[b] CNIT, S2N Lab, Genoa, Italy

ARTICLE INFO

Keywords:
6G
Value-driven
Sustainability
MDAF

ABSTRACT

In order to satisfy the ethical requirements that are expected from upcoming 6G technologies, the knowledge of the power consumption ascribable to the applications and network functions is crucial to enforce a sense of responsibility and joint efforts towards value-driven sustainability. In this respect, this paper presents a modular, energy-focused prototype of the Management Data Analytics Function (MDAF), that represents the cornerstone of the Observability framework recently developed by the 6Green Project. Its main goal is to provide management data to upper layers (i.e., network and end-users/verticals). A 5G network was tested on both the User-Plane and Control-Plane; at the same time, said network was monitored by both the proposed MDAF and a common power monitoring solution: Scaphandre. Results show that the MDAF measures a higher power consumption than Scaphandre; the difference lies between 50% (when idle) and 4% (at the maximum offered load). This difference corresponds to the "indirect" power consumption that the MDAF is able to ascribe to the containers/Virtual Machines (VMs) (i.e., the Network Functions (NFs)). The more accurate power consumption measurements of the single NFs is a first step towards the need to spread energy/carbon awareness to all the involved stakeholders.

1. Introduction

Even though every major technological innovation in history has contributed to shaping societies, the so-called Fourth Industrial Revolution [1] is not only transforming the means of production but pervading every aspect of our everyday lives. In this respect, mobile networks have played a pivotal role in closing the (geographical as much as economical) gap between disadvantaged communities and critical services such as health and education [2].

In this context, the coming of the 6th Generation of Mobile Technologies (6G) [3] is awaited with great expectations on the societal and economical values it will carry. In fact, even in these earliest stages of its definition, the intention is emerging to adopt a value-driven approach along with the traditional performance-driven perspective, and to juxtapose quantitatively measurable Key Performance Indicators (KPIs) with Key Value Indicators (KVIs) able to represent the value offered by the network and the ethical requirements expected from its stakeholders [4,5].

For instance, the focus on the sustainability of the upcoming technology [6] is being expanded from the environmental aspects to including economic and societal sustainability as well. One of the consequences on the enabling technologies that will be developed is that end-users will be seen as a full-fledged stakeholder of the 6G ecosystem, and as such will be provided with the tools [7] to make educated decisions on their KPIs, which will consequently impact on their own carbon footprint. While quantifying such an impact is far from an easy task, the path towards truly ethical networking cannot prescind from this knowledge.

Among the main issues preventing the assessment of the carbon footprint ascribable to a specific user, to an application, or to a slice, is due to the fact that their virtualized components most of the times run on the same piece of hardware, making it hard to understand to what extent a specific Virtual Machine (VM) or container is responsible for the power consumption of a server. Moreover, in a multi-domain ecosystem in which the infrastructure, the network platform and the vertical applications are owned by multiple separated entities [8], the already non-trivial task of inferring the consumption of a physical component ascribable to a virtual one, let alone share this information in a secure and trusted manner.

To this end, this paper presents a prototype of a Management Data Analytics Function (MDAF), designed and developed within the 6Green Project [9], which is part of a multi-domain observability framework introduced in [10].

The Observability Framework of 5/6G is composed of two main Network Functions (NFs): the Network Data Analytics Function (NWDAF)

---

* Corresponding author.
E-mail addresses: beatrice.siccardi@edu.unige.it, beatrice.siccardi@tnt-lab.unige.it (B. Siccardi).

https://doi.org/10.1016/j.comnet.2025.111487
Received 20 February 2025; Received in revised form 2 June 2025; Accepted 17 June 2025

and the MDAF. While several articles in the literature can be found reading the former, this is not the case for the latter. In the following, we report, to the best of our knowledge, the MDAF-focused articles currently available in the literature. The authors in [11] present a scaling and load-balancing mechanism based on the MDAF; in detail the proposed MDAF monitors the CPU utilization of the AMF instances, performs analytics on such data and thus takes a scaling or a load-balancing decision. [12] presents an End-to-End Data Analytics Framework for 5G Architecture of which the MDAF is part; their implementation is particularly focused on the Radio Access Network (RAN). In [13] data usage control mechanisms to safeguard private network data when performing management data analytics are examined. In the PoC, the MDAF's goal is to analyse energy savings. Finally, authors in [14] delve into the role the MDAF has in building historical databases leveraged by AI mechanisms for autonomous control, management, and orchestration in 5G.

The MDAF proposed in this article is focused on energy/power consumption: it represents an evolution to the 3GPP TS 28.533 definition [15], which already includes resource usage predictions, to extend the plethora of collected data to the energy context by interacting with the Infrastructure. Thanks to this interaction, the MDAF allows to quantify the portion of power consumed at bare-metal level ascribable to a VM or a container.

Results compare the MDAF and Scaphandre [16], one of the most widespread open-source tools for collecting process-level power, when evaluating the consumption of several NFs. Scaphandre was chosen because of its greater ability to keep track of the resource (i.e., CPU, memory and networking) utilization than one of its alternative: Kepler [17]. Finally, results show that the MDAF proves to be more accurate as it detects a higher power consumption than Scaphandre under all tested conditions.

The remaining of the paper is organized as follows. Section 2 explores the current issues in the assessment of sustainability and how observability is a cornerstone of ethical networking. Section 3 outlines the Observability framework proposed by 6Green, with the main component, the MDAF, presented in Section 4. Results are reported in Section 5, and conclusions are drawn in Section 6.

2. Observability: an enabler of value-driven sustainability

As defined by the Brundtland Commission in 1987, “Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs” [18]. Only in recent years, thanks to initiatives such as the UN Sustainable Development Goals (SDGs) [19] and the European Green Deal, industry and academia are more and more adopting a forward-looking, value-driven approach that accounts for the (positive or negative) impact that a technological innovation can have on current and future society.

Focusing on ICT technologies, so far most of the attention has been devoted to the reduction of GHG emissions and results have been underwhelming to say the least: since the first figures have emerged [20,21], it has been clear that 5G technologies have affected the Operational Expenditures (OpEx) of Mobile Network Operators (MNOs), sensibly increasing the energy consumption of the infrastructure. 5G has been the first generation of mobile communications to heavily rely on virtualization technologies. While such technologies have been considered intrinsically green, as they increase flexibility and scalability allowing to acquire/release networking and computing resources in an as-a-Service fashion, in reality the lack of a coordinated effort among the stakeholders acting on the physical infrastructure and the virtualized resources has prevented the proper application of power management schemes to the hardware devices, causing a waste of resources and, consequently, an increase in their carbon footprint and energy requirements.

Driven by the recent succession of geopolitical crises that have affected the entire World population, a renewed interest in addressing global and local societal challenges is finally rising. Being 6G the upcoming generation of mobile technologies, which have been the key innovation driver of society for a long time, it is of paramount importance to ensure that a value-driven approach is adopted since the earliest stages of its conception, not only because this technology is expected to be way more pervasive of our everyday lives than its predecessor and its footprint is potentially bigger, but also because of the ripple effect it may cause on the sustainability of other sectors. Research on the technological enablers required to improve network and device energy efficiency must be performed along with the sustainability of non-telecom sectors, for example, verticals can benefit from the adoption of 6G technologies by achieving better lifecycle management that in turn can lead to increased lifetime and reusability.

Even before diving into the actual design of a 6G architecture, a common vision on what an ethical and sustainable network should look like can be drawn by reflecting on the available body of knowledge on environmental sustainability. Lessons learned from previous research activities [22,23] point out that, unsurprisingly, keeping the network capacity at peak level 24/7 was not sustainable, but that unless the stakeholders relying on the infrastructure were made aware of the power saving mechanisms in place they would perceive them as malfunctions. This becomes even more relevant now that the players acting at the virtualized layers are directly responsible for the consumption experienced at the infrastructure: unless they are held accountable for the impact ascribable to their vertical applications or virtual network services, it will be impossible to achieve significant savings of resources and, consequently, energy.

An ethical network is not possible unless it is a fair network: where the as-a-Service paradigm has failed so far, is in its inability to truly identify the actual resource usage ascribable to a vertical application or a network slice. In fact, while Virtual Network Functions (VNFs) and application components are deployed within virtualized components (i.e., containers or pods) that are isolated among themselves, such components share the same platforms and physical resources; consequently, their actual contribution is the sum of the containers/pods energy consumption with all the energy devoted to the execution of all the processes needed to make the bare-metal server and the virtualized environments (i.e., a container management platform) work. Additional components, such as ventilation systems, heat management, cooling systems and so on could be also mapped onto the containers and the VMs. However, this would require the Infrastructure owner to deploy several probes and to provide data coming from such probes to the 5/6G network, and we did not consider this case in the following. It is not trivial to map this “indirect” energy consumption to containers/pods because physical resources are shared among them, but unless this information can be precisely inferred and used to ensure that each stakeholder relying on a physical infrastructure is held accountable for its share of carbon footprint, an ethical network characterized by a fair, direct proportionality between resource usage and billing will remain a delusion.

The framework presented in the next section is specifically designed to break down the energy and resource consumption ascribable to VMs and containers running on top of a physical infrastructure. This information, once processed, can then be used to feed analytics mechanisms (e.g., data mining and fusing) for generating KPIs/KVIs tailored to each stakeholder (even up to the end-user). For instance, a particularly somber forecasting on the consumption ascribable to a specific slice could motivate a vertical to ask for a new deployment that allows to save some energy, or the MDAF outcomes could be fed to other tools tailored to interface with the end-users, such as the one proposed by the EXIGENCE Project [24]. The goal is to spread awareness at all the layers and possibly suggest actions/compromises in order to improve the overall (from an environmental, economic and societal point of view) sustainability of the whole 6G continuum. Delving deeper, the

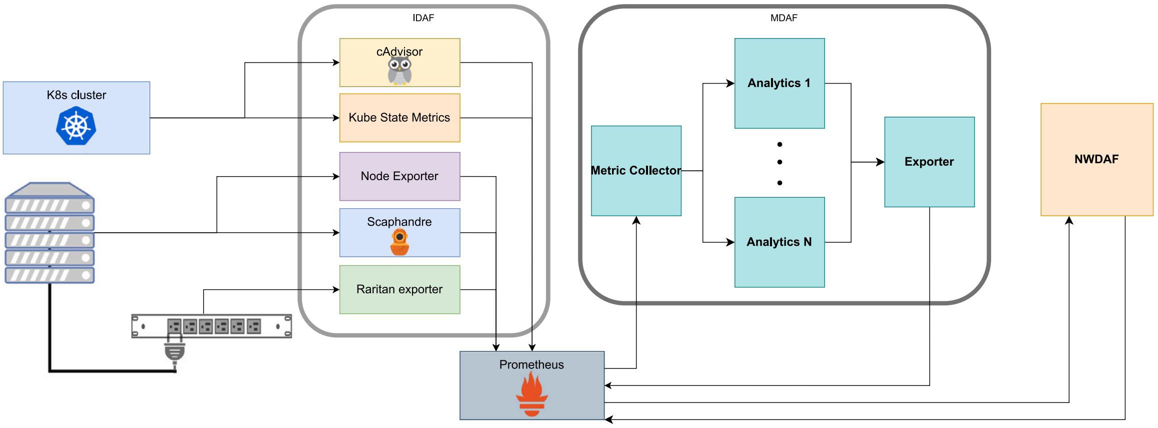

Fig. 1. Structure of the whole monitoring, observability and analytics platform ranging from the Infrastructure (IDAF) to the 5G Network domain (MDAF and NWDAF).

KPIs exposed to the final user would be the power consumption, and consequently the carbon emissions, for any action he/she performs on a mobile network (e.g., video streaming, web searches, online video games, etc.). This, on the one hand, raises awareness on the environmental impact, and, on the other hand, can be used in feedback loops that, through economic incentives, aim to save energy and decrease carbon emissions. In a nutshell, a user could be more prone to accept a decrease in quality while getting back both a practical incentive and a feedback highlighting how much power/carbon they saved with such a decision.

3. The observability framework

Although the focus of this paper is on the MDAF, it is important to present its interactions with the other functions in the Infrastructure and in the 5/6G Network domain. Therefore, the whole monitoring, observability and analytics platform is presented in Fig. 1. Each function will be explained in the following subsections.

3.1. The infrastructure data analytics function

The green block in Fig. 1 represents the Infrastructure Data Analytics Function (IDAF) which is responsible for exposing infrastructure-related metrics.

Since the 5/6G NFs should be deployed in the form of either VMs or containers, herein, the infrastructure concerns both the bare-metal servers and the container orchestrator platform (e.g., Kubernetes (K8s) in this case). In detail, we focus on containers rather than VMs, since the former is more flexible and “lighter” than the latter and currently preferred for deploying 5/6G NFs. However, it is worth highlighting from the start that our reasoning and also our development takes into consideration VMs as well.

Being the 5/6G Network Provider and the Infrastructure Provider usually different, it is essential to separate the data collection from the Infrastructure and from the Network. This reasoning leads to the separation between IDAF and MDAF. As a result, the MDAF will not have direct access to the infrastructure, but only to the metrics exposed by the IDAF. Additionally, this separation ensures that only the necessary data are available, and that the other, possibly sensitive, data is protected and not shared with the network.

As shown in Fig. 1, the IDAF is composed of five components. All of them are open-source exporters compatible exposing metrics in a Prometheus-like format. cAdvisor [25] and Kube State Metrics (KSM) [26] expose information regarding the Kubernetes cluster. The former can be either embedded in kubelet (i.e., the main K8s agent responsible for managing the deployment of pods and for the health of the K8s cluster) or installed as a docker container and exposes the resource (e.g., CPU, memory, networking, etc.) utilization per container. The latter is an additional service that exposes the state of K8s objects (e.g., pods, nodes, replicaset, statefulset, etc.).

NodeExporter [27] and Scaphandre [16] expose metrics of the bare-metal servers. The former exposes a wide variety of hardware- and kernel-related metrics among which it worth mentioning the resource (e.g., CPU, memory, networking, etc.) utilization of the servers. The latter exposes the power consumption of every single process (including containers and VMs) running on the server. Scaphandre relies on the Intel Running Average Power Limit (RAPL) [28] counters and on the time spent by each process on the CPU to compute the power consumption per process. So far, the power consumption provided by Scaphandre includes only the CPU and, only in some cases, the DRAM controller consumption. Therefore, being the CPU the most power-hungry component of servers, the metrics should be somewhat reliable for processes not needing GPUs.

Finally, the Raritan exporter [29] is in charge of exposing the power consumption of the Raritan Power Distribution Units (PDUs). In detail, we used the IX7™ PDU Controller by Raritan.

The aforementioned metrics are saved on a Prometheus [30] database which is common to all the three functions (i.e., IDAF, MDAF, NWDAF). Prometheus is a time series database collecting metrics composed of a timestamp, a value and optionally key–value labels. Since the Prometheus database will contain both insensitive and sensitive (e.g., concerning the UEs) data, controlling its access is of paramount importance. However, this topic is not discussed further in the paper because the sensitive data is produced by the NWDAF rather than the MDAF.

3.2. The management data analytics function

The MDAF, originally defined in 3GPP TS 28.53 [15], is a NF belonging to the 5G Service Based Architecture (SBA) in charge of collecting and performing analytics on management KPIs for NFs or network slices.

Being the proposed MDAF herein energy-efficiency focused, it is responsible for collecting power/energy KPIs coming from all the cloud-native infrastructure resources (i.e., bare-metal servers, VM hypervisor, container management platform). As highlighted above, the MDAF does not have direct access to the infrastructure in order to keep separation among tenants; but can rather access only the metrics exposed by the IDAF.

Moreover, the MDAF may use AI/ML technologies to perform analysis or predictions on the data. The MDAF will be explained thoroughly in the next section.

3.3. The network data analytics function

The NWDAF is responsible for the storage and data collection required for inference from the different NFs, as defined in the 3GPP TS 23.288. In particular, it collects additional analytics to allow inferring the MDAF preliminary mapping with control-level NF KPIs (e.g., the number and the type of Packet Data Unit (PDU) sessions, the distribution of UEs, the requests to each service exposed by NFs, etc.). It also computes KPIs through an analytics module and exposes the results to the NFs or external third-parties application. A model of NWDAF has been proposed by [31], which uses the Seasonal Auto-regressive Integrated Moving Average (SARIMA) model for data forecasting.

4. Description of the proposed MDAF

The MDAF is devoted to collect and manipulate management data. Consequently, at its level, we suppose not to have knowledge regarding the network, but rather only regarding the infrastructure. As mentioned earlier, the proposed MDAF is energy-efficiency focused, therefore the data mainly concern power consumption and resource utilization.

Due to virtualization, understanding the impact on power consumption of the single components of a 5G network is not trivial. In a nutshell, the MDAF obtains data from the IDAF and manipulates them to get the power consumption of the individual virtualized components (i.e., VMs and containers). As shown in Fig. 1, the MDAF is essentially composed of three blocks: the metric collector, the analytics blocks (which may be more than one) and the exporter.

Let us start by providing a general overview of the MDAF routine that is shown in Algorithm 1. First, some configuration parameters are loaded: the URL of the Prometheus instance and the port on which the exporter is going to expose the new metrics. Then, an infinite loop consisting of collecting metrics and applying analytics begins. It is worth noting that the exporter is initialized and started only after the first round of collecting metrics and performing analytics because it should start only after the metrics to be exposed are already available.

Algorithm 1 MDAF routine

```
prom_config=load_prom_config()
prom_manager=PrometheusManager(prom_config)

exp_config=load_exp_config()
exporter_initialized=False

while True:

    scrape_metrics()

    apply_analytics()

    if exporter_initialized==False
      exp=mdaf_exporter (exp_config)
      exp.start_exporter(exp_config)
      exporter_initialized=True
```

Moreover, it is worth mentioning that all the code was written in Python and that the MDAF was actually run as a K8s container to be coherent with actual 5G deployment.

In the following subsections, details on the MDAF components are provided. Eventually, in Section 4.4, the added value of our MDAF is highlighted.

4.1. The metric collector

The metric collector exploits the prometheus-api-client [32] library to interrogate the Prometheus instance. The only configuration needed concerns the address of Prometheus: IP:PORT. Reminding once again that the proposed MDAF is energy-efficiency focused, the metric collector is devoted to Scaphandre and cAdvisor metrics. Therefore, the main goal is to acquire the power consumption of each process and the resource utilization of each container.

It is of paramount importance to distinguish the processes based on types. Let us consider the set containing all the power per process metrics provided by Scaphandre: $S(t) = x_i(t) | i = 1, 2, \dots, N(t)$ where $N(t)$ is the number of processes in execution at time t.

From now on, we remove the time dependency for simplification purposes. To distinguish among the types of processes, we consider four non overlapping proper subsets of $S : K, D, V, E$ such that:

$$K \subset S, D \subset S, V \subset S, E \subset S$$

$$K \cup D \cup V \cup E = S$$

$$K \cap D = \emptyset, K \cap V = \emptyset, K \cap E = \emptyset, D \cap V = \emptyset, D \cap E = \emptyset, E \cap V = \emptyset$$

Then, each metric is assigned to one of the four subsets based on the following rule:

$$x_i = \begin{cases} x_i \in K | x_i & \text{is a K8s container} \\ x_i \in D | x_i & \text{is a Docker container} \\ x_i \in V | x_i & \text{is a virtual machine} \\ x_i \in E | x_i & \text{otherwise} \end{cases} \tag{1}$$

for $i = 1, 2, \dots, N$. From an implementation point of view, the selection of the correct subset is based on analysing the metric's labels.

Finally, it is worth deepening the role of subset E: it contains all the process' metrics that do not belong to virtualized components (i.e., containers (both K8s and Docker) and VMs). Therefore, subset E contains all the kernel-processes that are needed in order for the server, everything on top (hypervisor and the container orchestration platforms) to work correctly and for managing various operations such as reading/writing memory or handling packets. We assume that the servers would be off if not for the virtualized components (e.g., there may be a containerized 5G network running on K8s); hence the total power consumption of E represents the so-called "indirect" power consumption. The goal of the MDAF is be to assign the aforementioned power consumption to the virtualized components (i.e., to each element of the subsets K, D, V) in a balanced way.

4.2. Analytics

The Analytic block(s) are where the "intelligence" resides: its structure allows to plug in different algorithms depending on the current needs. In principle, herein any kind (i.e., possibly involving Artificial Intelligence as well) of data manipulation can be performed. These algorithms should be run online (i.e., while the 5/6G network is up and running and serving UEs) in order to both provide manipulated data upwards (e.g., to the final user) and to apply optimization decisions on the network (e.g., migration, scaling) in a live fashion.

In the proposed MDAF, only few simple operations are performed in order to finally compute the "indirect" power consumption of the virtualized components.

First, however, it is worth mentioning that an easier task concerning metric enrichment is performed. This is necessary to cope with a Scaphandre issue: it is not able to acquire labels for containers using containerd as container runtime interface (CRI). This, therefore, affects K8s containers which can still be uniquely identified through the container id, but do not have essential labels such as pod name and namespace. To solve this, we propose to merge the data coming from Kube State Metrics with the one from Scaphadre.

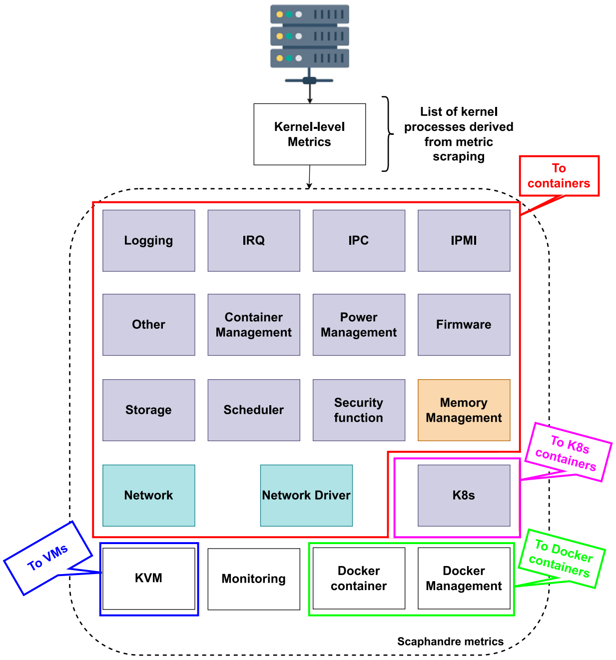

Fig. 2. Categorization of the kernel-level metrics (i.e., $\forall x_i \in E$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Then, focusing on the core purpose of the proposed MDAF, in the following we explain the algorithms we used to map the kernel-level power metrics mentioned in the previous subsection. First of all, we divide the $x_i \in E$ into categories. Such association is done “manually” thanks to a JSON file containing the associations between execution names and categories. This method was chosen to have a precise categorization. The chosen categories are 19 and are shown in Fig. 2. While the JSON file can and was updated after a thorough testing, another process classification mechanism could be considered to adapt the classification to a wide range of heterogeneous environments (i.e., featuring different kernel and background processes) and to the updates which any software application is subject to.

Once all the kernel processes are categorized, the total power consumption per category can be mapped onto the actual components of interest (i.e., containers and VMs).

Herein we focus on containers (both K8s and Docker) as the essential ones for the following reasons. First, because this way we can leverage on the resource utilization metrics provided by cAdvisor. Secondly, because containers are preferred when implementing a 5/6G network because they are lightweight (compared to VMs). Even though most 5/6G NFs are implemented on K8s since it offers a layer of orchestration (compared to Docker) and therefore makes instantiation and reconfiguration operations easier; we also keep a tight support for Docker containers since they are usually preferred for the containers needing multiple interfaces (i.e., the UPF).

The “indirect” power consumption is mapped differently according to the category as shown in Fig. 2. Most categories, the ones framed by the red, pink and green rectangles, are mapped to containers according to their percentage of resource utilization. An example of such algorithm for the CPU utilization is shown in Algorithm 2. This algorithm allows to ascribe more power to the most resource-hungry containers. As far as resources go, three are taken into account: CPU, memory and networking. The network and network management categories are associated to networking utilization (in particular to number of transmitted bytes), while the memory management one is associated to the memory utilization. The remaining categories are associated, in lack of an obvious link, to the CPU consumption, since said component is the most power-hungry in absence of GPUs. Then, the categories framed by the pink and green rectangles are mapped according to the CPU consumption to only the K8s or only to the Docker containers. It is worth mentioning that the processes boxed in the “Other” category are the ones which are difficult to relate to an operation/resource: for instance, the command line commands (e.g., sudo, grep, etc...) and the kworkers. The kworkers are processes that perform most of the actual processing for the kernel, especially in cases where there are interrupts, timers, I/O, etc. They may contain hints about their tasks in their execution names, however, it is generally difficult to gain insights about them.

The VM-related category (framed by the blue rectangle), in absence of resource utilization, is mapped in a uniform way to the VMs as shown in Algorithm 3.

Finally, the monitoring category is not mapped to any virtualized components, since it contains the power of monitoring processes that were not deployed as containers.

4.3. The exporter

The goal of the exporter is to expose the newly generated metrics in a Prometheus-like format. It leverages once again on the prometheus-api-client library to respect the Prometheus format. Moreover, the port on which it exposes the metrics is configurable.

An exhaustive list of the aforementioned metrics is provided in Table 1. Finally, it is worth mentioning that all of them are of Gauge type, that is a variable that can either decrease or increase, since they are representing power.

4.4. The added value of the proposed MDAF

The goal of this section is to highlight the added value of the proposed MDAF with respect to standard energy/power monitoring solutions (e.g., Scaphandre). In further detail, Scaphandre considers only the CPU power consumption (provided by Intel RAPL) and divides it, based on the CPU utilization, onto all the server processes. Thus, Scaphandre takes into account only the CPU utilization. Furthermore, it does not ascribe the “indirect” costs to VMs and containers. Clearly, considering only the CPU power consumption makes Scaphandre a good tool of comparison only in the case of CPU-intensive containers/VMs. While this is suitable for 5/6G NFs, it would be a great limitation in the case of GPU-intensive containers/VMs in which scenario another tool of comparison should be found.

Computing the power consumption of a single VM or a single container is not trivial at all. Our goal is to provide a comprehensive power consumption per container/VM that considers the “indirect” power consumption as well. Here lies the difference between the proposed MDAF and Scaphandre. Our rationale is based on the assumption that,

Algorithm 2 Mapping of kernel-level power according to resource utilization

```
$r_i$= Get CPU utilization per container from cAdvisor
total_cpu_utilization=0

$\forall i \in r_1$:
    total_cpu_utilization+=$r_i$

$\forall x_i \in K$:
    $\% = r_i$/total_cpu_utilization
    $x_i += \% \cdot$ category_power_cons
$\forall x_i \in D$:
    $\% = r_i$/total_cpu_utilization
    $x_i += \% \cdot$ category_power_cons
```

Table 1
Metrics exposed to Prometheus by the MDAF.

Metric name	Type	Labels	Description
mdaf_enriched_k8s_container_power_microwatts	Gauge	container_id, node, namespace, pod, cmdline, exe, pid	MDAF power consumption of K8s containers the additional labels (pod and namespace) which are missing from scaphandre (to cope with Scaphandre issues with containerd)
mdaf_aggregated_k8s_container_power_microwatts	Gauge	container_id, node, namespace, pod, cmdline, exe, pid	MDAF power (+ indirect) consumption of K8s containers
mdaf_aggregated_docker_container_power_microwatts	Gauge	container_id, container_names, container_scheduler, node, cmdline, exe, pid	MDAF power (+ indirect) consumption of docker containers
mdaf_aggregated_vm_power_microwatts	Gauge	vmname, node, cmdline, exe, pid	MDAF power (+ indirect) consumption of VMs
mdaf_category_power_microwatts	Gauge	node, category	MDAF power consumption of each category (composed of kernel-level metrics)
mdaf_monitoring_power_microwatts	Gauge	node	MDAF power consumption of the monitoring category (excluding docker containers). This metric will be empty unless you install some monitoring components directly as a service (e.g., Scaphandre)

Algorithm 3 Mapping of kernel-level power proportional to the number of VMs

```
$\forall x_i \in V$:
    $\% = 1/\#\_vms$
    $x_i += \% \cdot$ category_power_cons
```

in the absence of NFs, the servers hosting them would be turned off and consequently would be consuming zero power. Therefore, the goal of the proposed MDAF is to map all the power consumption of the servers to the NFs (i.e., VMs or containers).

Therefore, the "indirect" power consumption is mapped only to the processes of interest (i.e., mainly containers, but also VMs). In order to also take into account the other physical components utilization (not only CPU, but also memory and networking), the mapping is carried out, as explained in Section 4.2, through the CPU, memory, and networking utilization of the containers. Consequently, for instance, to the containers using more networking resources (sending and receiving more bytes) will be ascribed a bigger portion of the kernel-processes involving the networking area.

In a nutshell, the proposed MDAF provides a comprehensive power consumption of each NF; this is the first step to measure the power consumption of slices and end-users/vertical applications. Consequently, this begins the effort to address the need to spread awareness of the power consumption (which is, by definition, actually consumed at the infrastructure level) to all the stakeholders involved in the 5/6G ecosystem (from Infrastructure providers up to MNOs and end-users/verticals).

5. Performance evaluation

5.1. The experimental setup

The aim of the experiments reported in this section is to conduct measurements on a 5G network to show the usefulness of our proposed MDAF. In order to have a completed environment composed of both the core and the access network, we used an open-source 5G network Open Air Interface (OAI) [33] for the former, and a commercial product (i.e., LoadCore, [34]) for the latter. LoadCore is a 5G network simulator that makes use of "agents" to simulate a whole 5G network or only some components.

In further detail, the setup consists of two physical servers: one acts as the K8s master and the other one as the K8s worker. The master hosts most of the monitoring applications mentioned in Section 3.1 and all the containers needed to enhance the capabilities of Kubernetes (e.g., LoadBalancer, etc.). The K8s worker hosts the OAI 5G network under the form of K8s containers except for the UPF which is deployed as a docker container, the monitoring application needed running on both servers (i.e., Scaphandre and NodeExporter), the LoadCore components and the MDAF that, as mentioned earlier, is deployed as a K8s container since it would be part of the Control-Plane of the 5G network. Details of the full 5G network deployment are shown in Fig. 3. The white-coloured NFs are the ones provided externally (herein by OAI), the RAN (composed of gNodeB and User Equipments (UEs)) and the Data Network (DN) are simulated by LoadCore respectively by two agents deployed as VMs.

During the tests, both User- and Control-Plane operations were investigated. For both cases, data were collected from both Prometheus and LoadCore with a 3-s sampling period in the form of csv files.

For the User-Plane, UE(s) generating stateless UDP data flows were investigated. In further detail, independent UDP flows are generated from the UE(s) to the data network and vice versa. In this case, the offered load and the number of UEs were changed. First, we tested 1 UE generating a varying load from 1 kbps up to 100 Mbps (i.e., the maximum supported by OAI in this configuration). Then, we tested changing the number of UEs (1, 10 and 100) while keeping the aggregate offered load constant.

Regarding the Control-Plane, we tested two simple operations: UE attachment and PDU session establishment. In the former, we examined the situation in which 100 UEs are attaching with a 1 UE per second rate (i.e., the maximum allowed). In the latter, a single UE established PDU sessions with a rate of 1 per second; herein we have found the maximum number of PDU sessions handled by OAI to be equal to 85.

Finally, it is worth mentioning that each test was performed 10 times. Since no significant difference was noticed among the instances of the tests and since the temporal analysis has the most use, in the following only one instance for each test is presented.

5.2. User-plane

The first tests involved User-Plane traffic, as explained in the previous section, and allowed assessing the accuracy of the proposed MDAF prototype with respect to the state-of-the-art solution Scaphandre.

They involved measuring the power consumed by the UPF when User-Plane traffic is generated by one UE at varying rates, decreasing from 100 Mbit/s to 1 kbit/s. For each rate, traffic was transmitted for 15 min followed by 5 min of idle.

Fig. 4 reports the throughput obtained for the different offered loads. It can be seen that the OAI UPF is able to reach 100% throughput for traffic below 100 Mbit/s but experiences losses when the latter load is transmitted. This fact is only due to OAI limitations and has

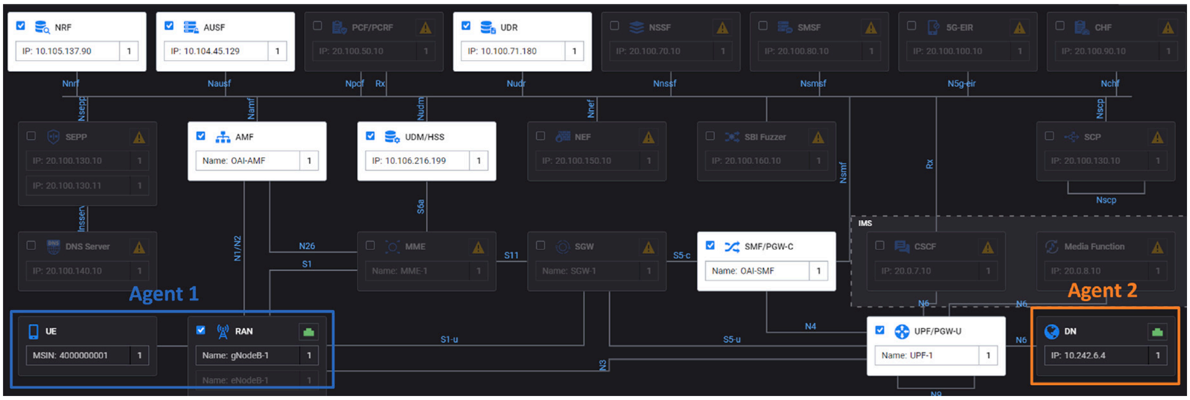

Fig. 3. Details of the test setup as shown by the LoadCore dashboard. The NFs in the white boxes are provided by OAI, while the components encased in the blue and orange rectangle are simulated by LoadCore respectively by two different agents. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

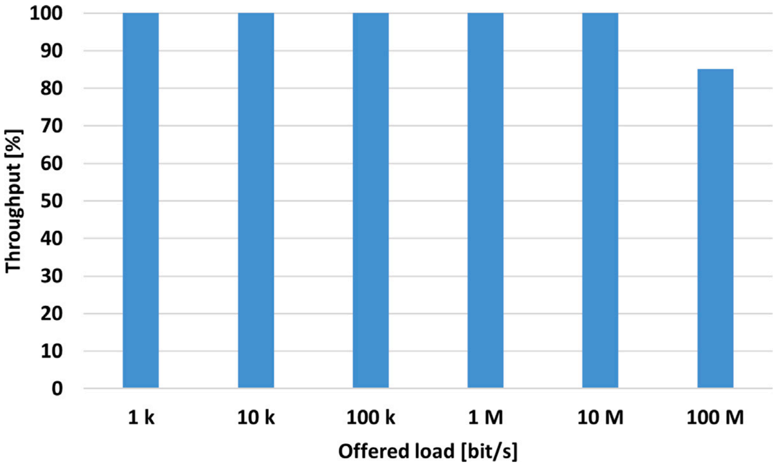

Fig. 4. Throughput obtained for the UPF under tests at varying offered load.

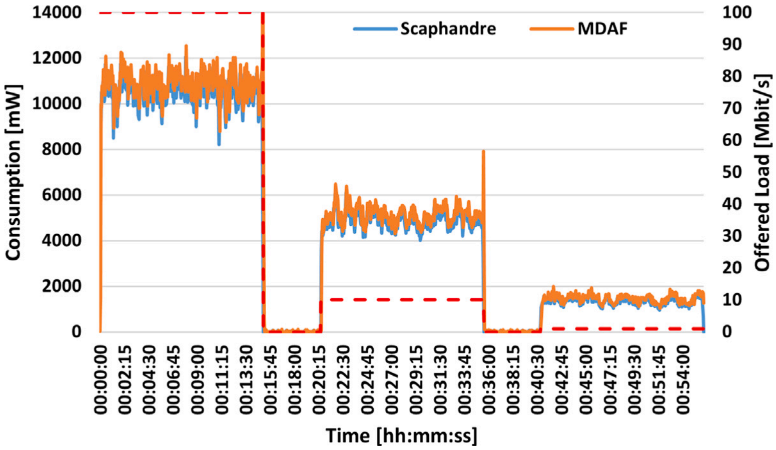

Fig. 5. Power consumption of the UPF measured for Scaphandre and MDAF during the test (part 1). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

no implications on our evaluation, but is worth reporting nevertheless for a coherent understanding of the following results.

Figs. 5 and 6 show the power consumption, at varying loads (represented with the red dotted line plotted on the secondary y axis), for both Scaphandre and MDAF. Although the general trend is similar, the graphs show a slightly lower consumption for the former. The difference between Scaphandre and MDAF is more evident by looking at Fig. 7, which reports the average power consumption for the different traffic loads and the percentage of the difference between the two tools, which is 50% in the absence of traffic, 33% for 1 kbit/s and going down as traffic grows to end slightly below 4% for the highest offered load.

Although the consumption not detected by Scaphandre is not very high (the average difference for 100 Mbit/s is around 400 mW), when this underestimation occurs for all the functions deployed in a network,

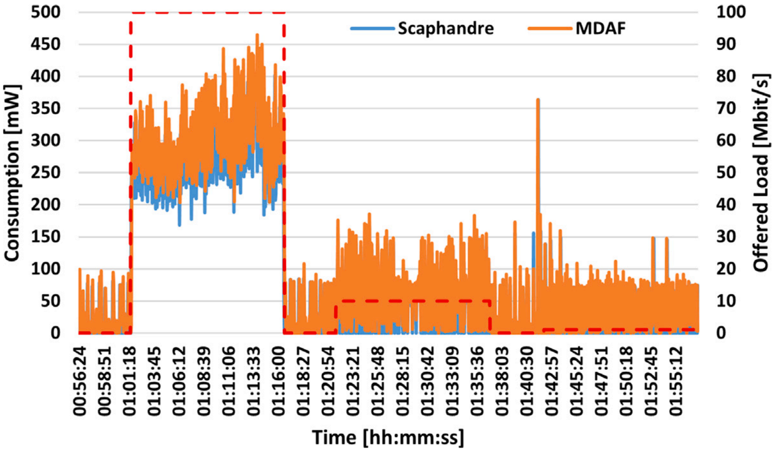

Fig. 6. Power consumption of the UPF measured for Scaphandre and MDAF during the test (part 2). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

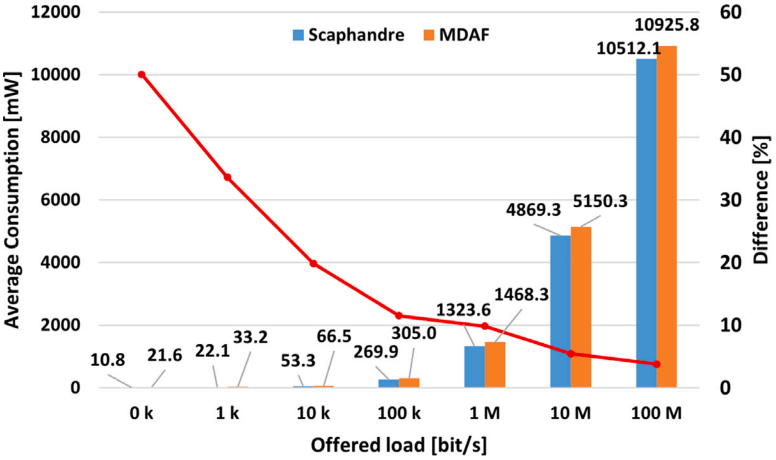

Fig. 7. Average consumption obtained for the UPF under tests at varying offered load.

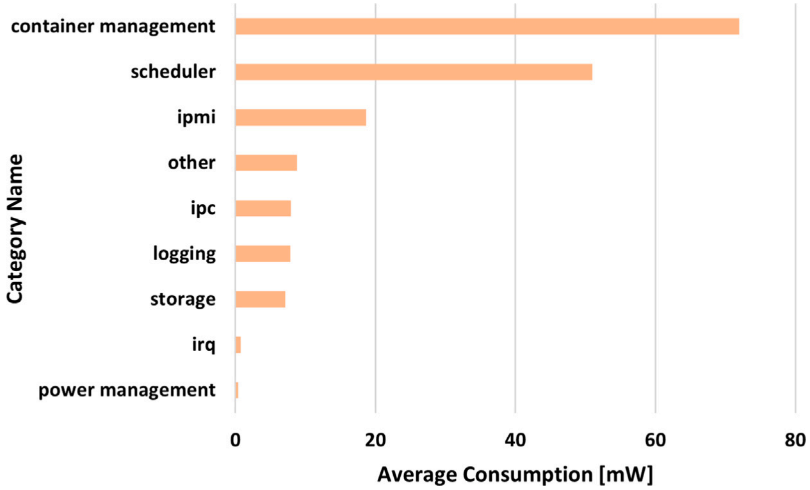

Fig. 8. Breakdown of the consumption ascribable to the UPF for the most relevant categories.

the impact can be severely critical and lead to misinformed resource allocations. This is especially true for the UPF that, being the sole User-Plane NF, is usually present in multiple instances in the network.

Reminding that the difference in the consumption measured by Scaphandre and MDAF is due to the "indirect" contribution of the kernel-level power metrics, it is possible to gain a better understanding of the data reported in Figs. 5 and 6 by taking a look into such metrics. As explained in Section 4.2, the consumption arising from the (categorized) kernel processes is assigned to a specific component according to its resource utilization (e.g., CPU, network and memory), a piece of information provided by the cAdvisor component in the IDAF. Fig. 8 reports the breakdown of the contribution to the UPF provided

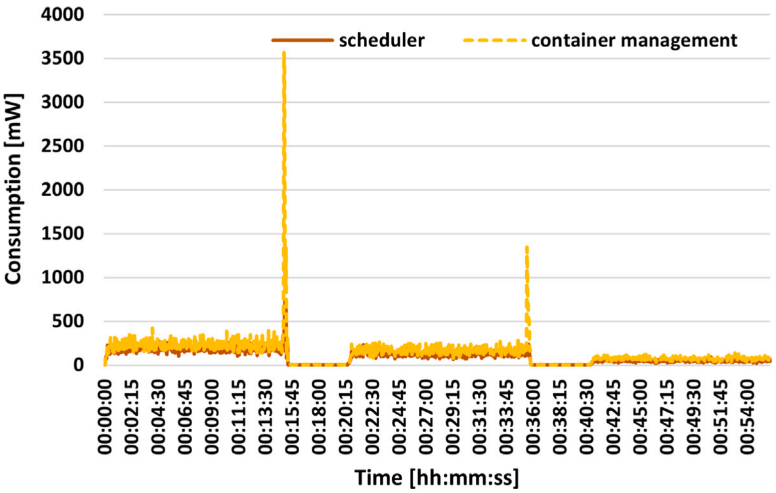

Fig. 9. Consumption of the “container management” and “scheduler” categories ascribed to the UPF.

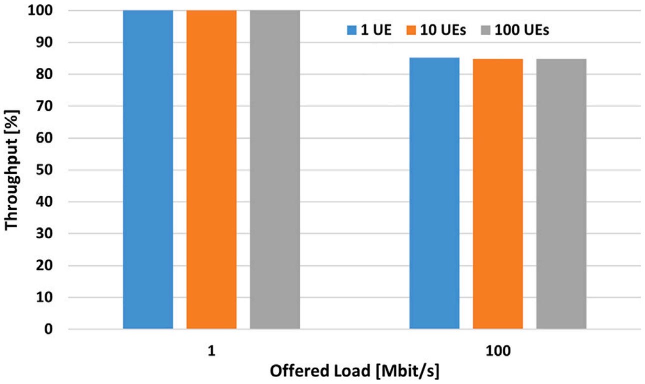

Fig. 10. Throughput obtained for the UPF under test at varying offered load and number of UEs.

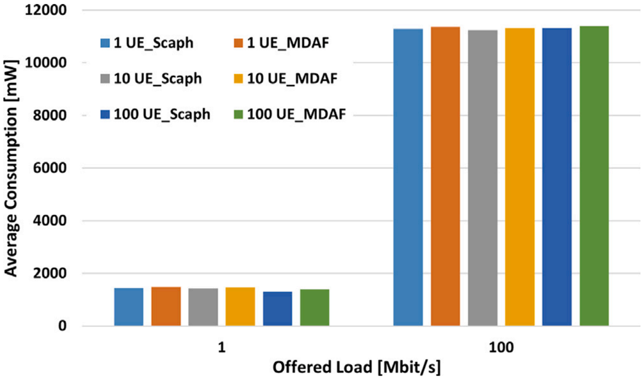

Fig. 11. Average consumption obtained for the UPF under test at varying offered load and number of UEs.

by the most power-hungry categories of processes. It is worth noting that all these categories are assigned according to the CPU utilization, which means that processes ascribed upon memory and network do not cause such a relevant overhead to the overall consumption of the UPF.

By looking again at Fig. 5, a spike in the consumption measured by the MDAF can be noticed when traffic transmission ends, which is particularly visible at 00:15:45 and 00:36:00, but is not detected by Scaphandre. By investigating the consumption ascribable to each category individually, it can be noticed that the spikes are mainly caused by the processes belonging to the “ContainerManagement” and “Scheduler” categories, which are the highest in Fig. 8. Fig. 9 reports their consumption obtained during the same test case as in Fig. 5, and one can see how their individual spikes account for 3500 and 1000 mW, respectively, which explains the 4000 mW spike from Fig. 5. The same consideration can be made for the second spike as well. Regardless of the reason for that surge in consumption, the proposed MDAF not only allows to detect it, but it also provides an outline of the processes involved to give the chance for proper investigation and identification of potential issues in the deployed instances.

An additional test was also carried out to measure the power consumed by the UPF in the presence of more than one UE. Namely, we tested two different loads, 1 Mbit/s and 100 Mbit/s, but this time the load was obtained by aggregating the traffic generated by 1, 10 and 100 UEs.

Figs. 10 and 11 report the throughput and the power consumption obtained at varying traffic and number of UEs generating it. In the User-Plane, it can be seen that no differences appear when transmitting traffic from one or more than one UE so, for the sake of brevity, for the remaining of the paper we only considered the case of one UE. On the contrary, the higher number of UEs causes an overhead on the Control-Plane due to UE registrations and PDU session establishments. However, this is not reported because, as shown in the following Section, the power consumption of the Control-Plane is much lower in amplitude and lasts for a very short time compared to the User-Plane; as a result, its contribution is negligible.

Finally, it is worth discussing what would happen to the User-Plane results in different scenarios. First of all, the type of traffic can be analysed. Herein, only UDP was chosen since it is the simplest type of traffic and it can model well some 5/6G applications (i.e., video streaming). Another possible choice could have been TCP: a reliable transport protocol that, through acknowledgements, retransmits lost packets. In this case, a difference would be detected in the highest load case (100 Mbit/s): the lost packets would be retransmitted; therefore, the power consumption and the throughput would increase. In case of a mixed type of traffic (i.e., UDP and TCP), all of the above would still hold. Another change in scenario can involve the regularity of the traffic. For instance, in order to model a real 5/6G network traffic, the offered load would need to change frequently during the day and it would have a peak during the most busy hours while having a fall during the least busy ones. The irregular traffic load would result in several changes in the resource utilization of the UPF. This would possibly change the assignment of the physical resources to the UPF container and, consequently, the overall power would differ due to the variation in the “sleep” states to which the CPUs are assigned.

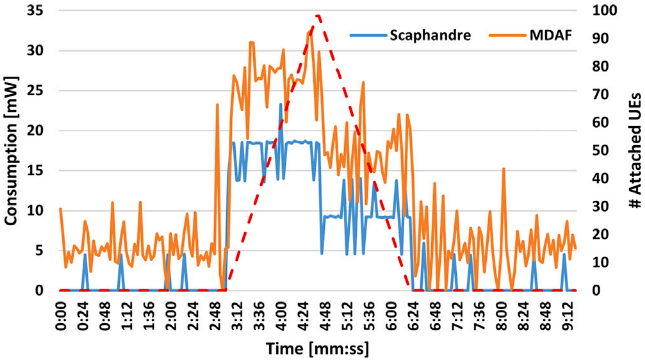

Fig. 12. Power consumption of the AMF measured for Scaphandre and MDAF during UE attachment operations.

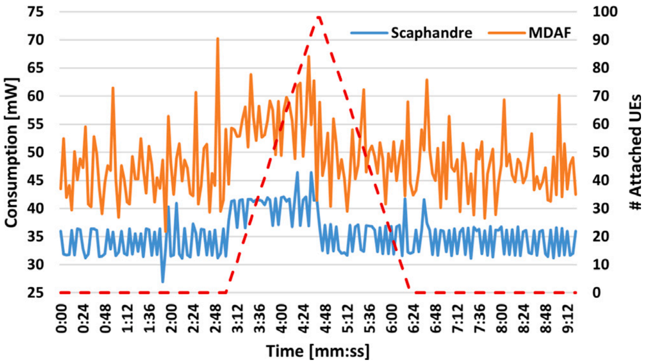

Fig. 13. Power consumption of the AUSF measured for Scaphandre and MDAF during UE attachment operations.

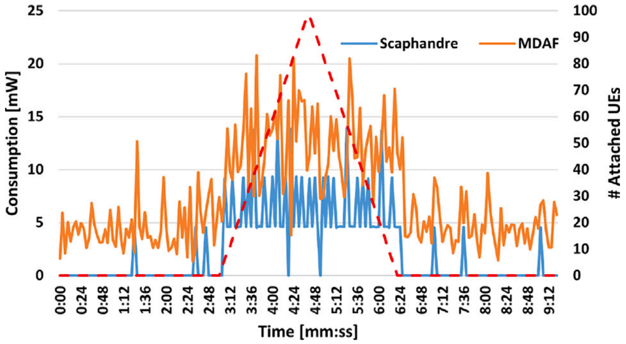

Fig. 14. Power consumption of the SMF measured for Scaphandre and MDAF during UE attachment operations.

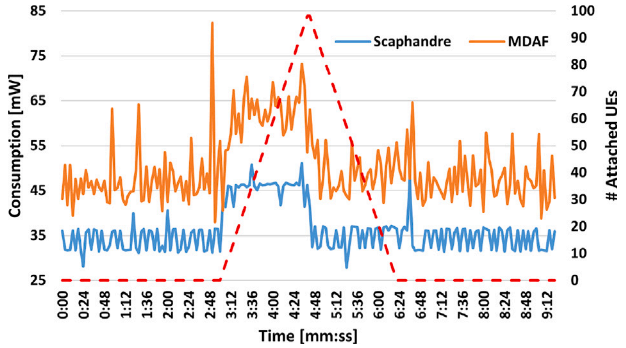

Fig. 15. Power consumption of the UDM measured for Scaphandre and MDAF during UE attachment operations.

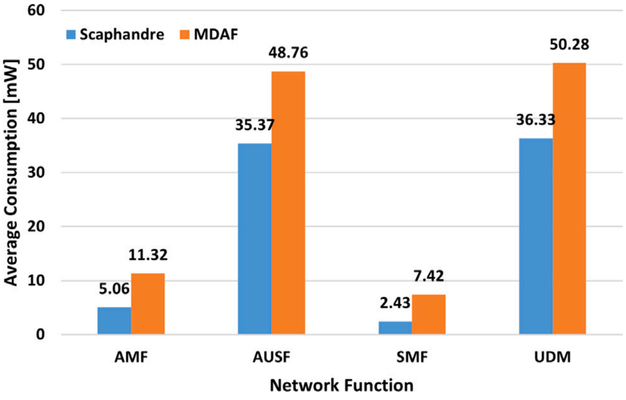

Fig. 16. Average consumption obtained for the Control-Plane functions during UE attachment operations.

5.3. Control-plane

For the evaluation of the consumption ascribable to the Control-Plane, we chose two operations, namely UE attachment and PDU session establishment, and measured the consumption of several NFs involved in these operations with the two tools. Namely, the Access and mobility Management Function (AMF), Authentication Server Function (AUSF), Session Management Function (SMF) and Unified Data Management (UDM) have been investigated.

The selection has been made because these functions perform a plethora of operations that occur upon diverse triggers: the AMF is responsible for all the signalling related to user authentication and mobility, including non-access stratum protocols; the AUSF further provides the support for the authentication of untrusted non-3GPP access; the SMF is responsible for the lifecycle management of PDU sessions; finally, the UDM is a database that stores runtime information used by the AMF. For both UE attachment and PDU creation, tests lasted around nine minutes and we collected the power consumed by each considered NF.

In the first test, LoadCore was used to simulate the attachment and de-attachment of 100 UEs, one at a second. Figs. 12–15 show the consumption for the AMF, AUSF, SMF, and UDM, respectively, measured with Scaphandre and MDAF, along with the number of attached UEs, reported with the dotted line.

For all of these functions, we can recognize the same trend happening for the UPF, albeit at a smaller scale: while Scaphandre allows to detect the changes due to the attachment and detachment of the UEs, which is reflected by an increase of the power consumption at 3:00 and a decrease at 4:45 when the UEs begin detaching, the inability to account for the “indirect” consumption results in an underestimation that on average varies, depending on the NF, between 5 and 14 mW, as reported in Fig. 16. In this figure, it can be noticed that, while AMF and SMF overall consume less, the consumption detected with the MDAF is more than double the amount reported by Scaphandre. Moreover, it can be noted that, in the absence of attached UEs, Scaphandre only detects some repetitive consumptions that correspond to the “keep-alive heartbeats” towards the Network Repository Function (NRF).

The second test considered the creation of a growing number of PDU sessions up to 85, which is the current limit for OAI, their maintenance for 36 s, and their release. Figs. 17–20 show the consumption of the same NFs as in the previous case, and similar considerations can be drawn. Here, the main difference is that AUSF and UDM do not show any significant variations when PDU creation/destruction operations

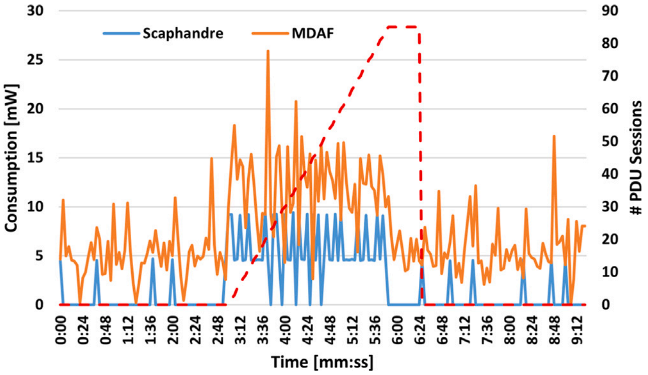

Fig. 17. Power consumption of the AMF measured for Scaphandre and MDAF during PDU session lifecycle operations.

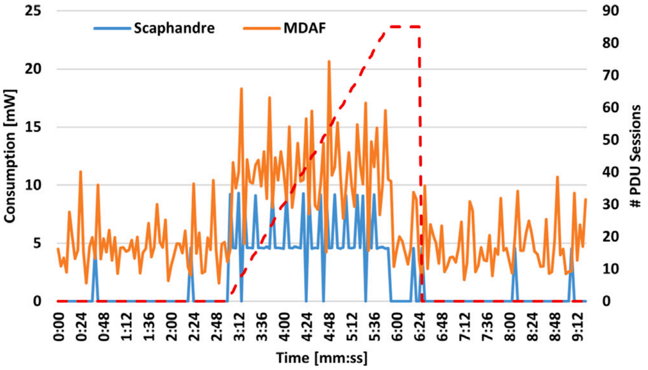

Fig. 19. Power consumption of the SMF measured for Scaphandre and MDAF during PDU session lifecycle operations.

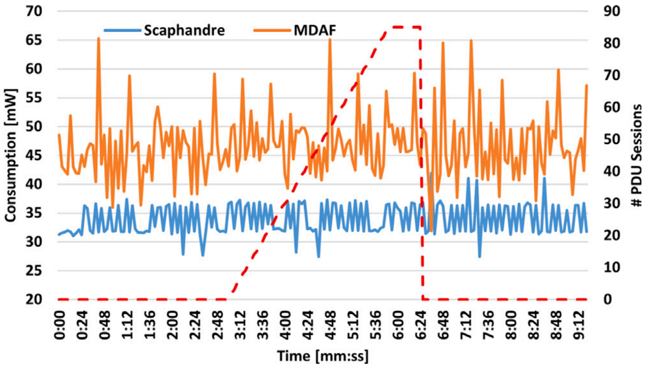

Fig. 18. Power consumption of the AUSF measured for Scaphandre and MDAF during PDU session lifecycle operations.

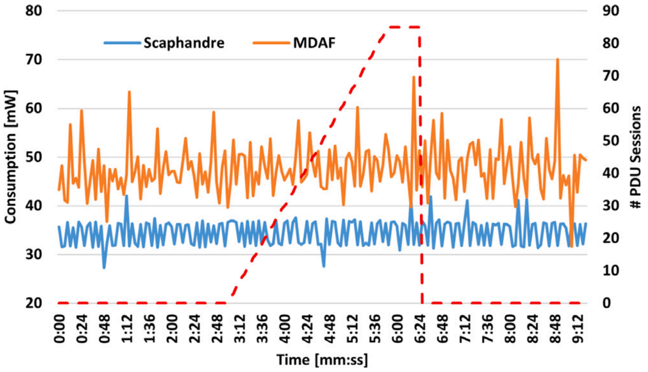

Fig. 20. Power consumption of the UDM measured for Scaphandre and MDAF during PDU session lifecycle operations.

occur. This can be easily explained by the fact that the AUSF and UDM are involved in the UE authentication when it attaches to the network; therefore they are "stressed" only at the beginning of the PDU session establishment test. Moreover, it can be seen that the consumption measured by Scaphandre for these two NFs is not only around 13 mW lower with respect to the MDAF (see Fig. 21), but it also presents less variations. Focusing on the AMF and SMF, the same behaviour in the absence of operations on the PDU sessions as in the previous test appears; additionally, the difference between the consumptions measured with Scaphandre and MDAF grows, emphasizing a stronger impact on CPU operations that Scaphandre alone is not able to detect.

6. Conclusions

In recent years, a paradigm shift is steering the conversation on sustainability towards a broader meaning that includes economic, societal and environmental aspects thanks to a value-driven approach that aims to combine quantitative KPIs with value metrics aiming to represent significant ethical principles. We believe that a clear understanding of the footprint ascribable to the virtual instances relying on a physical infrastructure is crucial to ensure responsibility and realize a truly ethical network.

In this paper we presented a power-focused MDAF which is a crucial block of the 6G Observability framework. The MDAF is the NF that can bridge the gap between the infrastructure and the network layer which are usually owned by different entities. It can infer to upper layers (network and end-users/verticals) management information coming from the infrastructure. The MDAF we propose herein has a modular structure that allows to add other analytics modules (potentially exploiting AI) further down the line.

Tests were performed on both the User-Plane and the Control-Plane of the network, and the MDAF accuracy was compared with a very common power monitoring tool: Scaphandre.

Overall, results show that the MDAF measures a higher power consumption than Scaphandre. This is due to fact that we take into account also the "indirect" power consumption of each virtualized entities (i.e., containers and VMs). The difference ranges from a minimum of 4% (with a 100 Mbit/s offered load) to a maximum of 50% (when no traffic is generated). It is also worth mentioning that the MDAF is able to detect peaks that Scaphandre does not. Results show that the "indirect" power consumption is mainly ascribable to the kernel-processes related to "Container Management" and "Scheduler" operations.

The additional power may not seem too high in the results shown here, however it is crucial to highlight that, in real conditions, the NFs would have multiple instances (this is especially true for the UPF) and therefore the impact of the "indirect" power would, in fact, be much higher.

The proposed MDAF generates more accurate power measurements per NF. This is the first step required to spread an energy/power awareness to upper stakeholders. Thus, we provide information at the single NF level of granularity; the future works include the level just above: the network slice. These measurements presented in this paper can be aggregated in order to estimate the power consumption of each 5/6G network slice.

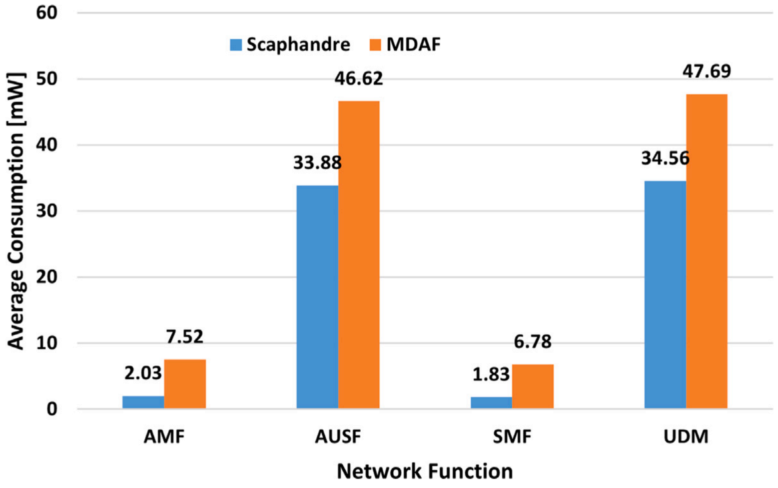

Fig. 21. Average consumption obtained for the CP functions during PDU session lifecycle operations.

CRediT authorship contribution statement

Milad Akbari: Writing – original draft, Software. **Raffaele Bolla:** Writing – original draft, Software. **Roberto Bruschi:** Writing – original draft, Software. **Chiara Lombardo:** Writing – original draft, Software. **Nicole Simone Martinelli:** Writing – original draft, Software. **Beatrice Siccardi:** Writing – original draft, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been partially supported by the Horizon Europe Research and Innovation Action 6Green (Grant Agreement no. 101096925) and by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP F83C22001690001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART") in the Net4Future project.

Data availability

Data will be made available on request.

References

[1] The fourth industrial revolution: What it means, how to respond. World economic forum, 2025, available at https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/. (Accessed 30 May 2025).

[2] Mobile money transaction volume in africa in 2020 and 2021, by region, 2025, available at https://www.statista.com/statistics/1139403/mobile-money-transactions-africa/. (Accessed 30 May 2025).

[3] Position paper - key strategies for 6G smart networks and services, 2025, available at https://6g-ia.eu/wp-content/uploads/2023/10/6g-ia-position-paper_2023_final.pdf. (Accessed 30 May 2025).

[4] What societal values will 6G address? Societal key values and key value indicators analysed through 6G use cases, 2025, available at https://5g-ppp.eu/wp-content/uploads/2022/05/What-societal-values-will-6G-address-White-Paper-v1.0-final.pdf. (Accessed 30 May 2025).

[5] G. Wikström, N. Bledow, M. Matinmikko-Blue, H. Breuer, C. Costa, G. Darzanos, A. Gavras, T. Hossfeld, I. Mesogiti, K. Petersen, P. Porambage, R.-A. Stoica, S. Wunderer, Key value indicators: A framework for values-driven next-generation ict solutions, Telecommun. Policy 48 (6) (2024) 102778, http://dx.doi.org/10.1016/j.telpol.2024.102778, URL https://www.sciencedirect.com/science/article/pii/S0308596124000752.

[6] White paper - sustainability of 6G: Ways to reduce energy consumption, 2025, available at https://6g-ia.eu/wp-content/uploads/2025/01/sustainability_of_6g-path_forward_v1.2.2.pdf. (Accessed 30 May 2025).

[7] CENTRIC - towards an AI-native, user-centric air interface for 6g networks, 2025, available at https://centric-sns.eu/. (Accessed 30 May 2025).

[8] R. Bruschi, R. Bolla, F. Davoli, A. Zafeiropoulos, P. Gouvas, Mobile edge vertical computing over 5G network sliced infrastructures: An insight into integration approaches, IEEE Commun. Mag. 57 (7) (2019) 78–84, http://dx.doi.org/10.1109/MCOM.2019.1800425.

[9] 6Green - green technologies for 5/6G service-based architectures, 2025, available at https://www.6green.eu/. (Accessed 30 May 2025).

[10] M. Akbari, R. Bolla, R. Bruschi, C. Lombardo, N.S. Martinelli, B. Siccardi, Observe to sustain – how to enable beyond 5G networks to target sustainability goals, in: 2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2024, pp. 1–7, http://dx.doi.org/10.1109/PIMRC59610.2024.10817341.

[11] C. Hsiung, F.J. Lin, J.-C. Chen, C. Chen, 5G network slice scalability based on management data analytics function (MDAF), in: S.-Y. Hsieh, L.-J. Hung, R. Klasing, C.-W. Lee, S.-L. Peng (Eds.), New Trends in Computer Technologies and Applications, Springer Nature Singapore, Singapore, 2022, pp. 587–598.

[12] E. Pateromichelakis, F. Moggio, C. Mannweiler, P. Arnold, M. Shariat, M. Einhaus, Q. Wei, Ö. Bulakci, A. De Domenico, End-to-end data analytics framework for 5G architecture, IEEE Access 7 (2019) 40295–40312, http://dx.doi.org/10.1109/ACCESS.2019.2902984.

[13] H. Zafar, U. Fattore, F. Cirillo, C.J. Bernardos, Data usage control for privacy-enhanced network analytics in private 5g networks, IEEE Open J. Commun. Soc. (2024) http://dx.doi.org/10.1109/OJCOMS.2024.3522379, 1–1.

[14] D. Bega, M. Gramaglia, R. Perez, M. Fiore, A. Banchs, X. Costa-Pérez, AI-based autonomous control, management, and orchestration in 5G: From standards to algorithms, IEEE Netw. 34 (6) (2020) 14–20, http://dx.doi.org/10.1109/MNET.001.2000047.

[15] 5G; management and orchestration; concepts, use cases and requirements (3GPP TS 28.530 version 16.2.0 release 16), 2025, available at https://www.etsi.org/deliver/etsi_ts/128500_128599/128530/16.02.00_60/ts_128530v160200p.pdf. (Accessed 30 May 2025).

[16] Scaphandre, 2025, available at https://github.com/hubblo-org/scaphandre. (Accessed 30 May 2025).

[17] M. Akbari, R. Bolla, R. Bruschi, F. Davoli, C. Lombardo, B. Siccardi, A monitoring, observability and analytics framework to improve the sustainability of B5G technologies, in: 2024 IEEE International Conference on Communications Workshops (ICC Workshops), 2024, pp. 969–975, http://dx.doi.org/10.1109/ICCWorkshops59551.2024.10615948.

[18] "Report of the world commission on environment and development: Our common future", world commission on environment and development (WCED), 2025, available at https://sustainabledevelopment.un.org/content/documents/5987our-common-future.pdf. (Accessed 30 May 2025).

[19] United nations, 2020, the 17 goals, 2025, available at https://sdgs.un.org/goals. (Accessed 30 May 2025).

[20] "Energy efficiency: An overview," GSMA future networks, 8th May 2019, 2025, available at https://www.gsma.com/futurenetworks/wiki/energy-efficiency-2/. (Accessed 30 May 2025).

[21] J. Lorincz, A. Capone, J. Wu, Energy-efficient and sustainable networks: State-of-the-art and new trends, Sensors 19 (2019) 4864.

[22] R. Bolla, R. Bruschi, F. Cucchietti, F. Davoli, Setting the course for a green internet, Science 342 (2013) 1316.

[23] R. Bolla, R. Bruschi, F. Cucchietti, F. Davoli, Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures, IEEE Commun. Surv. Tutor. 13 (2011) 223–244.

[24] EXIGENCE integrates measurement, optimisation and incentivisation, 2025, available at https://projectexigence.eu/. (Accessed 30 May 2025).

[25] cAdvisor, 2025, available at https://github.com/google/cadvisor. (Accessed 30 May 2025).

[26] Kube state metrics, 2025, available at https://github.com/kubernetes/kube-state-metrics. (Accessed 30 May 2025).

[27] Node exporter, 2025, available at https://github.com/prometheus/node_exporter. (Accessed 30 May 2025).

[28] K. Khan, M. Hirki, T. Niemi, J. Nurminen, B. Schölkopf, Z. Ou, RAPL in action: Experiences in using RAPL for power measurements, ACM Trans. Model. Perform. Eval. Comput. Syst. 3 (2023) 1–26.

[29] Raritan PDU exporter, 2025, available at https://github.com/psyinfra/prometheus-raritan-pdu-exporter. (Accessed 30 May 2025).

[30] Prometheus, 2025, available at https://prometheus.io/. (Accessed 30 May 2025).

[31] R. Bolla, P. Bono, R. Bruschi, C. Lombardo, N.S. Martinelli, B. Siccardi, An open-source prototype of network data analytics function for next-generation 5/6G environments, in: 2023 IEEE Globecom Workshops (GC Wkshps), Kuala Lumpur, Malaysia, 2023, pp. 720–725.

[32] prometheus-api-client, 2025, available at https://pypi.org/project/prometheus-api-client/. (Accessed 30 May 2025).

[33] OPENAIR-CN-5G: An implementation of the 5G core network by the OpenAir-Interface community, 2025, available at https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/tree/master. (Accessed 30 May 2025).

[34] LoadCore - core-network-solutions, 2025, available at https://www.keysight.com/it/en/product/P8900S/loadcore-corenetwork-solutions.html. (Accessed 30 May 2025).