GREEN 2024

The Eighth International Conference on Green Communications, Computing and Technologies

ISBN: 978-1-68558-203-6

November 3rd - 7th, 2024

Nice, France

GREEN 2024 Editors

Lorena Parra, Universidad Politécnica de Madrid, Spain

GREEN 2024

Forward

The Ninth International Conference on Green Communications, Computing and Technologies (GREEN 2024), held on on November 3-7, 2024, continues the series of events focusing on current solutions, stringent requirements for further development, and evaluations of potential directions. The event targets are bringing together academia, research institutes, and industries working towards green solutions.

Expected economic, environmental and society wellbeing impact of green computing and communications technologies led to important research and solutions achievements in recent years. Environmental sustainability, high-energy efficiency, diversity of energy sources, renewable energy resources contributed to new paradigms and technologies for green computing and communication.

Economic metrics and social acceptability are still under scrutiny, despite the fact that many solutions, technologies and products are available. Deployment at large scale and a long term evaluation of benefits are under way in different areas where dedicated solutions are applied.

We take here the opportunity to warmly thank all the members of the GREEN 2024 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to GREEN 2024. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the GREEN 2024 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that GREEN 2024 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of green communications, computing and technologies. We also hope that Nice provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

GREEN 2024 Chairs

GREEN 2024 Steering Committee
Daniele Codetta-Raiteri, Università del Piemonte Orientale, Italy
Sanjeev Sondur, Oracle / Temple University, USA
Mamadou Baïlo Camara, University of Le Havre Normandy, France
Bernard Lee, HedgeSPA, Singapore

GREEN 2023 Publicity Chair
Lorena Parra Boronat, Universitat Politecnica de Valencia, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain
Francisco Javier Díaz Blasco, Universitat Politecnica de Valencia, Spain
Ali Ahmad, Universitat Politecnica de Valencia, Spain

GREEN 2024

Committee

GREEN 2024 Steering Committee
Daniele Codetta-Raiteri, Università del Piemonte Orientale, Italy
Sanjeev Sondur, Oracle / Temple University, USA
Mamadou Baïlo Camara, University of Le Havre Normandy, France
Bernard Lee, HedgeSPA, Singapore

GREEN 2024 Publicity Chair
Lorena Parra Boronat, Universitat Politecnica de Valencia, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain
Francisco Javier Díaz Blasco, Universitat Politecnica de Valencia, Spain
Ali Ahmad, Universitat Politecnica de Valencia, Spain

GREEN 2024 Technical Program Committee

Khelifa Abdelkrim, Unité de Recherche Appliquée en Energies Renouvelables - URAER | Centre de Développement des Energies Renouvelables - CDER, Ghardaïa, Algeria
Kouzou Abdellah, Ziane Achour University of Djelfa, Algeria
Ali Ahaitouf, USMBA, Morocco
Salem Al-Agtash, German Jordanian University, Jordan
Daniel Albiero, Universidade Estadual de Campinas (UNICAMP), Brazil
Angelo Algieri, University of Calabria, Italy
Mahmoud Amin, Manhattan College, USA
Zacharoula Andreopoulou, Aristotle University of Thessaloniki, Greece
Hala Alami Aroussi, Mohamed Premier University, Morocco
Carlos Alberto Astudillo Trujillo, State University of Campinas, USA
Abdellah Bah, ENSAM | Mohammed V University in Rabat, Morocco
Figen Balo, Firat University, Turkey
Hajji Bekkay, ENSA-Oujda | Mohammed First University, Morocco
Rachid Benchrifa, Université Mohammed V, Morocco
Lazhar Benmebrouk, Ouargla University, Algeria
Suman Bhunia, Miami University, Ohio, USA
Anne Blavette, SATIE (CNRS) | ENS Rennes | Univ. Rennes, France
Guillaume Bourgeois, La Rochelle Université, France
Khalid Bouziane, International University of Rabat, Morocco
Mamadou Baïlo Camara, University of Le Havre Normandy, France
M. Girish Chandra, TCS Research Whitefield, India
Dana Ciupageanu, National University of Science and Technology POLITEHNICA Bucharest, Romania
Daniele Codetta-Raiteri, Università del Piemonte Orientale, Italy
Luigi Costanzo, Università degli Studi della Campania Luigi Vanvitelli, Italy
Naouel Daouas, National Engineering School of Monastir, Tunisia
Rekioua Djamila, University of Bejaia, Algeria
Rachid El Bachtiri, USMBA University, Fez, Morocco

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Containerization's Power Use Overhead in Video Streaming

Etienne-Victor Depasquale
Department of Communications and Computer Engineering
University of Malta
Msida, Malta
e-mail: edepa@ieee.org

Saviour Zammit
Department of Communications and Computer Engineering
University of Malta
Msida, Malta
e-mail: saviour.zammit@um.edu.mt

Abstract— **Containerization of a service enables live migration and, thereby, consolidation of running service instances onto as few host platforms as possible. However, containerization's operational overhead must be investigated to determine overall viability. One dimension of this overhead is that of power use, and this is investigated here. An architecture for a video cache service at the edge of a Communications Service Provider's (CSP) network in the metropolitan area is designed, and a scaled version is implemented in a laboratory environment. A comparison is made between power used while streaming videos in both native and containerized modes of operation. Containerization is found to incur a low power overhead while streaming video, compared with streaming video from ffmpeg running directly on the host operating system. Power use is measured using hardware instrumentation and with PowerTOP, a software power meter. Limits on the latter's accuracy have been observed.**

Keywords - containers; power; video; streaming; implementation model.

I. INTRODUCTION

Content Delivery Networks (CDNs) are overlay networks that are key to controlling the growth in demand for bandwidth in long-haul communications links. By distributing content to caches in geographical regions of the world where customers are located, the number of times which a single item of content crosses long-haul links between the content origin's region and the customer's region, is reduced to just one. In turn, the content is distributed several times to customers in the region. While the function of the CDN, from the customer's perspective, is that of reducing latency and avoiding buffer underrun, the control of bandwidth growth is a function that has a strategic role in the stability of world-wide communication. The CDN's role in bandwidth control continues to gain attention [1]; a variety of CDN implementations has been investigated [2][3] and surveyed [4][5] and generalized surveys are of ongoing interest [6][7]. The importance of the CDN seems to grant sufficient ground for study of the impact of its point of presence (PoP) on the information and communication technology of its environs.

This study seeks to compare power use in containerized deployment of the media server in a CDN PoP. It focuses on the power use of the media server as it processes a representative set of tasks. The media selected for study is video (henceforth, the media server will be referred to as the video server), and two reasons support this choice. Video dominates traffic, whether in the access, aggregation, metro-core, or long-haul. Moreover, some of the tasks, such as transcoding, are processor-intensive and serve to indicate the power capacity required to support CDN PoPs.

The rest of this paper is structured as follows. In Section II, the objective is stated. In Section III, the implementation model is presented. This supports reproduction of the test environment. In Section IV, the method is elaborated upon. Section V presents the results and Section VI supports interpretation through analysis of these results. Section VII draws a succinct conclusion on the impact that containerization of a video service has on power use overhead.

II. OBJECTIVE

An overhead is expected in the containerized implementation, and its ***quantification*** is sought. The objective can be articulated in terms of a comparison between two types of deployment:

- power use in a computer system that runs the service within containers, with
- power use in a computer system that runs the service directly on the operating system.

Quantification is sought in order to control a tradeoff between native and containerized deployment. The tradeoff may be succinctly summarized as one of greater operating power per unit (physical host) versus potential for lower number of operating units (physical hosts). The following sections elaborate on this summary.

III. IMPLEMENTATION MODEL

An edge cache of a video streaming service is deployed. A high-level view of the implementational model is shown in Figures 1 and 2.

- Figure 1 shows an implementation that is easily portable to a cloud-native infrastructure (henceforth referred to as the cloud-native implementation), and
- Figure 2 shows an implementation that is a hybrid of physical (the video server) and virtual network functions (the switch).

The cloud-native implementation uses containers to host the video server. Both implementations host a virtual layer 2 switch in the intermediate node.

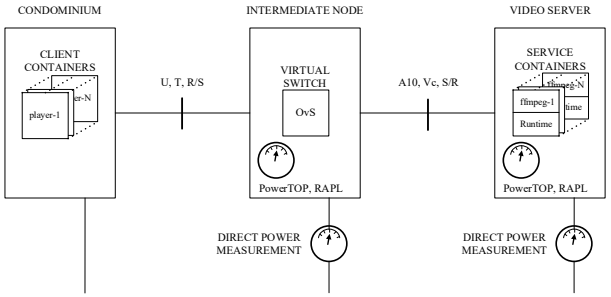

Figure 1. Physical topology of the video streaming service, deployed in containers. Video Server located in local exchange or Access Node (AN); Intermediate Note located in street cabinet (subtended AN [8]).

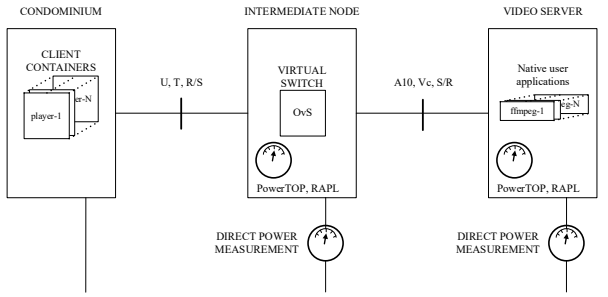

Figure 2. Physical topology of the video streaming service, deployed on a host operating system.

A. Hardware

The hardware used in this testbed consists of a set of three HPE (Hewlett Packard Enterprise) ProLiant BL460c Gen9 blade servers [9], hosted in an HPE c7000 blade enclosure. Connectivity between server and client blades is obtained through pass-through interconnect bay modules, patched with single-mode optic fibre cables. These latter modules support the goal of bypassing c7000 ecosystem interconnect-bay physical networking devices. Bypass is necessary to introduce separate, **virtual** switching hardware. The virtual switch is implemented on a third HPE Gen9 blade server. The links to the switch are of type 10GBASE-SR. The video server has a single Intel® Xeon® CPU E5-2640 v3 (2.60GHz) processor package. Dynamic Voltage and Frequency Scaling (DVFS) is under system firmware control.

B. Software

The software consists of:

- an FFmpeg [10] video server. This is representative of the access node at the edge of the metro-core network;
- a TSDuck [11] receiver. This is representative of end-user's video player, and is also used to measure received bitrate to ensure that Quality of Service (QoS) (see Section IV-C) is respected;
- the virtual switch software is Open vSwitch [12].

A minimalist operating system was selected for the video server, to support isolation and attribution in power measurements. While minimalist operating systems do not necessarily correlate with minimal noise in power measurement, it seems useful to reduce the number of possible sources from the outset. For this reason, Alpine Linux [13] Standard distribution version 3.19 was chosen.

The container system software selected is Docker [14]. Docker is a mature containerization platform and it is modular: the runtime daemon (containerd) supports other user interfaces apart from the Docker user interface (dockerd). For example, Kubernetes [15] can be used to manage containers created through the Docker Command-Line Interface (CLI).

IV. METHOD

A. Instrumentation

Near-real time measurement of power use can be obtained from two sources of instrumentation. The blade servers are equipped with a management processor (known as "integrated lights-out", or iLO) that logs a power measurement every 10 seconds and stores a 20-minute history that can be read through a Redfish®[16] - compliant RESTful (Representational State Transfer) Application Programming Interface (API). Selectivity in aggregate power use measurement is afforded by blade systems, since these separate power supply to the (blade) computer system from power supply to two major overhead power drains. Blade servers use blade chassis services for power supply (where ac – dc conversion losses occur) and cooling (where blowers use power as they ventilate from chassis front to chassis rear). Thus, measurement of power used by the blade server at the supply voltage rails is free of the problematic, variable contribution from overheads, and idle power can be measured to the accuracy afforded by these blade system power measurement instruments. The measurement datum is of integer type, obtained by truncation of the decimal part of the actual measurement. Moreover: since the iLO is not part of the System Under Test (SUT), it does not alter power measurement.

While the iLO provides an aggregate power measurement, process- and thread- level granularity is obtained through software power meters. Hardware extensions for power measurement are available in processor models that support the HoweverIntel Running Average Power Limit (RAPL) feature. PowerTOP [17] is software that enables this level of power attribution, and it is indeed capable of exploiting RAPL. This tool complements the aggregate power measurement obtained by blade sensor instrumentation. PowerTOP uses a top-down approach [18], (it divides the power measurement over a period amongst processes and threads in proportion to their core utilization) and precedes the measurement period by one of calibration (the utility was run in calibration mode for several hours before starting the first experiment) in which it obtains weighting parameters for the attribution process. Calibration is further refined with use, and PowerTOP saves its parametric refinement to persistent storage for future exploitation [19]. PowerTOP was used in its logging mode of operation, with 10 (ten) – second

averaging intervals. However, PowerTOP has several significant limitations, as follows: it only measures dynamic power, it does not capture all power use, and it increases the SUT's aggregate power use. These must be mitigated.

B. Baselining

It is necessary to distinguish power used by the video service from power used by other consumers. This requires measurement of static (/idle/leakage) power use. It is also necessary to distinguish between dynamic power used during video service operation time, from dynamic power used when the service is idle. In essence: service power use can be thought of as an amount added above that used by the operating system and system software, which in turn is added above that used to operate electronic components (leakage/static/idle) power. Hence, it is possible to perceive a baseline to which service power is added to obtain the total power. Formally:

$$P_{b_1}^{(video)} = P_{idle}^{f_1} + P_q^{(os)}$$

where $P_q^{(os)}$ is the **dynamic** power corresponding to the Operating System's (OS) operation **without** container system software and without running User Applications (UAs), and $P_{idle}^{f_1}$ is the **idle/leakage/static** power at the frequency f_1 at which the OS is quiescent.

A second baseline, $P_{b_2}^{(video)}$, is required to ensure experimental reproducibility.

$$P_{b_2}^{(video)} = P_{idle}^{f_2} + P_q^{(os+dockerd+containerd)}$$

Here, $P_q^{(os+dockerd+containerd)}$ is the **dynamic** power corresponding to the OS's operation **with** container system software but without running User Applications (UAs), and $P_{idle}^{f_2}$ is the **idle/leakage/static** power at the frequency f_2 at which the Operating System (OS) is quiescent. The state of quiescence is defined below (see IV-D-2).

C. Quality of Service

QoS is considered to be satisfied as long as there is sufficient capacity in the links to keep the overall average received bitrate of every video stream at or above the video file's overall bitrate.

D. Experiments

1) Test conditions

Video service will be delivered from both containerized and native deployments. The test conditions pertinent to the video server will be the following.

1. Implementation
   a. During containerized operation, each video service process and the libraries on which it depends will be operated from a container. One service process serves one client.
   b. During native operation, a new instance of the video service process will be started for every new client.
2. Load unit: This will consist of the work required to process a workflow based upon a video with the following technical specifications:
   a. Overall bitrate = 457 kb/s, = video bitrate of 326 kb/s + audio bitrate of 127 kb/s + mp4 container metadata rate (overhead)
   b. Duration = 1h 32m 2.19s (5522.19 s), of which 30 minutes are played, starting at a randomly-selected point in the video.
   c. H.264 video codec, Main profile
      i. Resolution = 1280 x 720
      ii. Frame rate ≈ 23.98 frames/second (fps)
   d. Advanced Audio Coding (AAC) audio codec, Low Complexity profile
      i. Sampling rate = 44.1 kHz
   e. Client supports same video and audio codec; hence server does not need real-time transcoding.

2) Procedure

The power used by the video server is measured at progressively higher load levels. Two sets of experiments are carried out: the first set uses containerized video server instances and the second set uses native video server instances. A containerized service instance consists of a container carrying ffmpeg. A single container is created to deliver a single stream and is destroyed immediately thereafter. When the container is created, ffmpeg is executed and listens on a TCP port, through which it streams 30 minutes of video. A native service instance is a single instance of the ffmpeg process; it follows the same lifecycle as the containerized instance.

Management of operations is not trivial, even at the minimum load level, as it involves the following steps:

1. Reboot the video server, to obtain a common and reproducible initial state.
2. Wait until the video server quiesces. This is the time required for server power use to fall to the state where the iLO measurement persistently shows baseline 2 usage. Persistence was empirically found to be ascertained 20 minutes after rebooting.
3. Start the power meters for both total and dynamic power, for both the video server and the virtual switch.
4. Wait for a fifteen-minute interval, to capture behaviour before video streaming.
5. Instantiate and start a container carrying the ffmpeg listener, poised for real-time playback with randomized starting point and 30-minute play time.
6. Start a TSDuck client to connect to the container and measure the bitrate, averaged over 5-second intervals.
7. Once 30 minutes of video have been played, destroy the container.
8. Wait for a fifteen-minute interval, to capture behaviour after video streaming.

For several concurrent streams, steps 5 and 6 must be repeated for each one of the additional streams. For the native service instance, step 5 involves the ffmpeg process only and there is no equivalent to step 7.

It seems evident that manual management is highly prone to error and is therefore unsuitable. Automated management using Python scripts and Ansible [20] is employed to handle

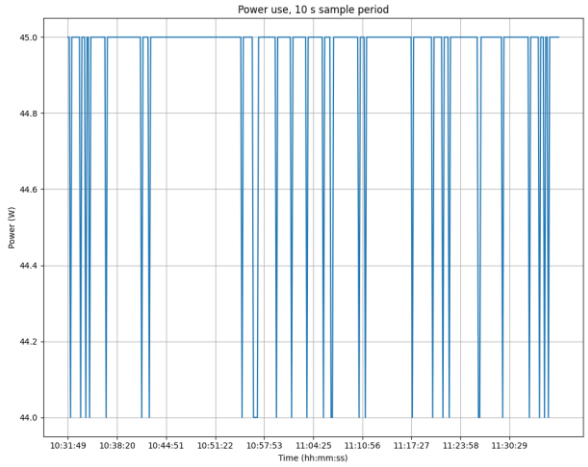

Figure 3. Power used by the video server, with a quiescent OS.

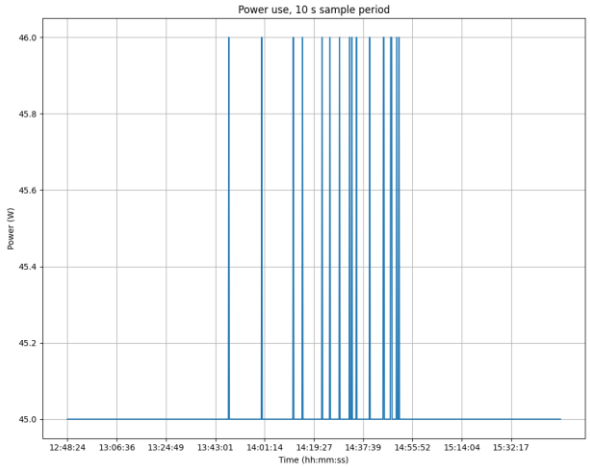

Figure 4. Baseline 2 video server aggregate power.

the **orchestration** of the various roles: power meters, container runtime managers and video clients. This enables the experiment to be scaled out to levels that are well beyond the physical limitations of a single human operator.

V. RESULTS

Denote:

- mean dynamic power measured by PowerTOP by $\overline{p_{dyn}^{(ptop)}}$
- mean total power measured by the iLO during a time period T_x by $\overline{p^{(iLO)}}([T_x])$.

A. Video server's baseline 1

Figure 3 shows the power used by the video server over an hour period of measurement, post-onset of quiescence. Since the iLO truncates decimals in $[n, n+1)$ to n, then the computation of the mean will count the incidences of 45 W and 44 W, and use them as weights to compute a lower limit to the range of values which the average can take. An upper limit is obtained by adding the maximum possible error (equal to 1W) and the mean of the possible range obtained by adding the mean error (0.5W) to the lower limit of the range. Using this premise, the mean power measured by the iLO, under the condition of a quiescent operating system (see Figure 3) is as follows:

$$P_{idle}^{f_1} + P_q^{(os)} = \overline{p^{(iLO)}}([10{:}31{:}49, 11{:}37{:}01]) = 45.4198\ \text{W} \cong 45.4\ \text{W}$$

B. Mitigation of PowerTOP's limitations

PowerTOP captures neither static nor dynamic power used by Hard Disk Drives (HDDs) and Solid-State Disks (SSDs); this was observed and confirmed through discussion with PowerTOP's developers [21]. Indeed, our experiments under baseline 1 conditions show that if PowerTOP is operated in logging mode with HDD as destination, iLO aggregate power use is more than 0.5 W greater on the SUT than the figure obtained while logging to a RAM disk. While logging to RAM disk (under baseline 1 conditions), average aggregate power use increases to 45.5W, compared with 45.4W (see Section V-A, above) when measurements are taken solely through use of the iLO's instrumentation.

C. Video server's baseline 2

The difference in average dynamic power is added to baseline 1, to obtain baseline 2:

$$\Delta\overline{p_{dyn}^{(ptop)}} = \overline{p_{dyn}^{(ptop)}}(baseline\_2) - \overline{p_{dyn}^{(ptop)}}(baseline\_1) = 0.7727 - 0.1851 = 0.5876\ \text{W}$$

$$\therefore P_{b_2}^{(video)} = P_{idle}^{f_2} + P_q^{(os+dockerd+containerd)} = P_{idle}^{f_1} + P_q^{(os)} + \Delta\overline{p_{dyn}^{(ptop)}} = 45.4 + 0.5876 \cong 45.99\ \text{W}$$

This is consistent with the graphical summarization of iLO measurements shown in Figure 4. This baseline, notably the graph of power against time, is essential in obtaining a reproducible starting state for all video service operation experiments.

D. Orchestration of containerized streaming

Results from running experiments on 1, 2, 5, 10, 20, 40 and 80 instances are presented. The result items consist of:

1. Mean aggregate power use (iLO instrumentation). Due to the integer type of the measurement, actual average iLO power use can lie in the range of ± 0.5 W of the reported result.
2. Mean dynamic power use (PowerTOP instrumentation). Dynamic power data is added to baseline 1 and the sum is plotted on the same Cartesian axes as the total power data.

PowerTOP was used to attribute dynamic power to processes, and these were sorted in descending order. Graphical representations of the power used were produced too. These results are presented in the Github online repository at [22]. Measurements of received stream bitrates are also available in this repository.

1) Single instance

Table I shows the mean power use; Figure 5 shows PowerTOP's measurements offset by baseline 1 and laid over the iLO's measurements. Time is shown in the format hh:mm:ss, where hh, mm and ss stand for hour-of-day, minutes in the hour and seconds in the minute, respectively. The larger post-operation (post-op) average power is due to activity undertaken by an instance of containerd (the container runtime) after the container is destroyed (post-ops). However, well after operations end, the iLO's measurements return to the baseline 2 profile. Pre-operations (pre-ops), both meters (iLO and PowerTOP) are in good agreement (PowerTOP's measurements would all be rounded down to 45W). Moreover, the average power used during operations as estimated by PowerTOP is 46.99 W (baseline_1, = 45.4, + 1.5940), whereas the iLO estimates 47.03W. The ten-second averages' dissimilarity increases during and post-operations but is still good. Notably, the spike in power use at the beginning and end of operations is captured by both meters, albeit not being measurements of the same magnitude.

2) Two instances

Table II and Figure 6 show the results pertinent to two containerized video server instances. As is the case with the single instance, for pre-ops and post-ops, both meters are in good agreement (the spike at about 09:29:00 is probably due to HDD input/output operations while loading PowerTOP). During operations, the average total power estimated by PowerTOP is 48.02 W (baseline_1 + 2.6162), whereas the iLO estimates 47.06W. The discrepancy is an overestimate by about 1W.

An interpretation of the discrepancy between operating period averages is visible in the graph (Figure 6) showing real time measurements. When the iLO measures 46W, the actual value is in the range [46,47), and the rate of change between 46W and 47W is much larger than the single-instance case.

TABLE I. MEAN POWER USE – SINGLE SERVICE INSTANCE

Power type	Description	Avg[a] (W)
$\overline{p^{(iLO)}}[14:47:05,15:03:00]$	Before starting the service instance	45.65
$\overline{p^{(iLO)}}[15:03:00,15:33:05]$	During the service instance's operation	47.03
$\overline{p^{(iLO)}}[15:33:05,15:52:17]$	After the service instance ended	46.17
$\overline{p_{dyn}^{(ptop)}}[14:48:17,15:03:00]$	Mean dynamic power before service instance operation	0.8593
$\overline{p_{dyn}^{(ptop)}}[15:03:00,15:33:05]$	Mean dynamic power during service instance operation	1.5940

a. Average.

PowerTOP's real time measurements are consistently higher than 47W, revealing that several of the 10-second measurement intervals are in certain disagreement, albeit small (< 2/46, i.e., < 5%).

3) Five, ten, twenty, forty and eighty instances

The results for five (Table III, Figure 7), ten (Table IV, Figure 8), twenty (Table V, Figure 9), forty (Table VI, Figure 10) and eighty instances (Table VII, Figure 11) are shown below. Conditions pre-operations are similar, but PowerTOP's average error estimation increases as power use increases. The numbers shown in the list are PowerTOP's estimate vs iLO's maximum estimate, for N instances (Ni):

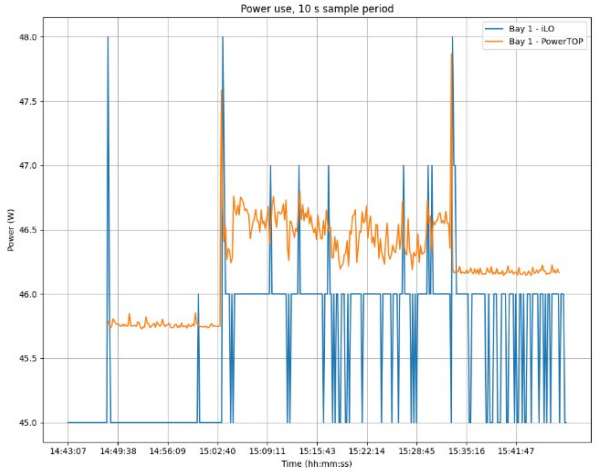

Figure 5. One instance. Video server's power use during containerized service operation. Baseline 1 added to powertop measurements.

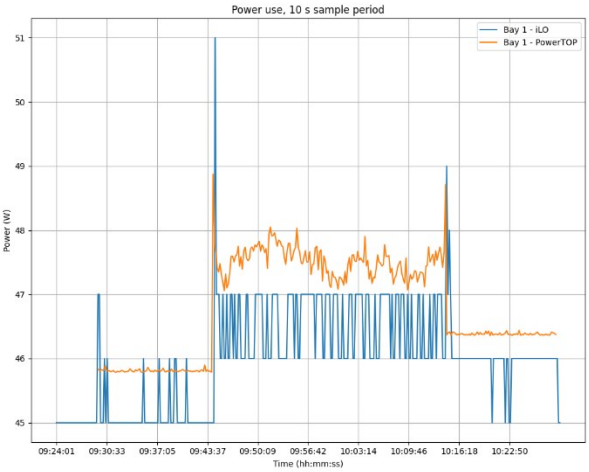

Figure 6. Two instances. Video server's power use during containerized service operation. Baseline 1 added to powertop measurements.

TABLE II. MEAN POWER USE – TWO SERVICE INSTANCES

Power type	Description	Avg (W)
$\overline{p^{(iLO)}}[09:24:01,09:44:27]$	Before starting the service instance	45.60
$\overline{p^{(iLO)}}[09:44:27,10:14:37]$	During the service instance's operation	47.06
$\overline{p^{(iLO)}}[10:14:37,10:29:23]$	After the service instance ended	46.12
$\overline{p_{dyn}^{(ptop)}}[09:29:27,09:44:27]$	Mean dynamic power before the service instances' operation	0.9693
$\overline{p_{dyn}^{(ptop)}}[09:44:27,10:14:37]$	Mean dynamic power during the service instances' operation	2.6162

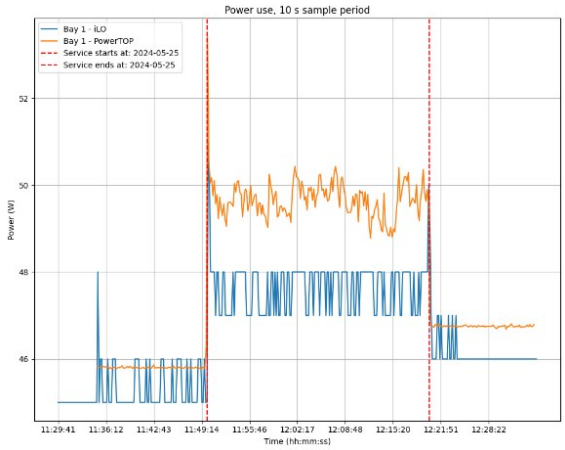

Figure 7. Five instances, containerized operations, baseline 1.

- 5i: 50.14 vs 48.66W
- 10i: 54.38 vs 50.10W
- 20i: 60.77 vs 51.74W
- 40i: 64.59 vs 53.90W
- 80i: 60.92 vs 56.80W

Inspection of the online supplementary data on process – level power attribution suggests that PowerTOP overestimates across all processes on our test platform.

TABLE III. MEAN POWER USE – FIVE SERVICE INSTANCES

Power type	Description	Avg, (W)
$\overline{p^{(iLO)}}[11:29:41,11:49:59]$	Before starting the service instance	45.79
$\overline{p^{(iLO)}}[11:49:59,12:20:15]$	During the service instance's operation	48.16
$\overline{p_{dyn}^{(ptop)}}[11:35:00,11:49:59]$	Mean dynamic power before service instances' operation	0.9970
$\overline{p_{dyn}^{(ptop)}}[11:49:59,12:20:15]$	Mean dynamic power during the service instances' operation	4.7421

TABLE IV. MEAN POWER USE – TEN SERVICE INSTANCES

Power type	Description	Avg. (W)
$\overline{p^{(iLO)}}[15:06:49,15:27:03]$	Before starting the service instance	45.60
$\overline{p^{(iLO)}}[15:27:03,15:57:27]$	During the service instance's operation	49.60
$\overline{p_{dyn}^{(ptop)}}[15:12:03,15:27:03]$	Mean dynamic power before service instances' operation	0.8759
$\overline{p_{dyn}^{(ptop)}}[15:27:03,15:57:27]$	Mean dynamic power during the service instances' operation	8.9781

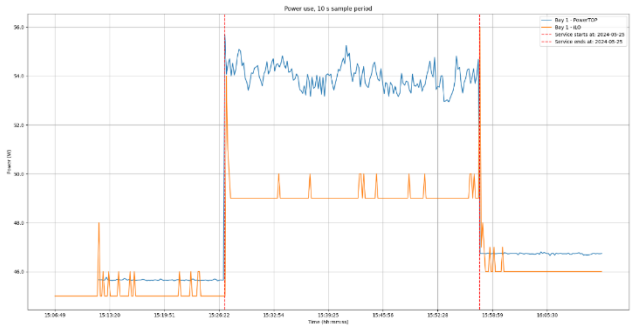

Figure 8. Ten instances, containerized operations, baseline 1.

TABLE V. MEAN POWER USE – TWENTY SERVICE INSTANCES

Power type	Description	Avg. (W)
$\overline{p^{(iLO)}}[17:56:58,18:17:08]$	Before starting the service instance	45.76
$\overline{p^{(iLO)}}[18:17:08,18:48:00]$	During the service instance's operation	51.24
$\overline{p_{dyn}^{(ptop)}}[18:02:08,18:17:08]$	Mean dynamic power before service instances' operation	0.8913
$\overline{p_{dyn}^{(ptop)}}[18:17:08,18:48:00]$	Mean dynamic power during the service instances' operation	15.3720

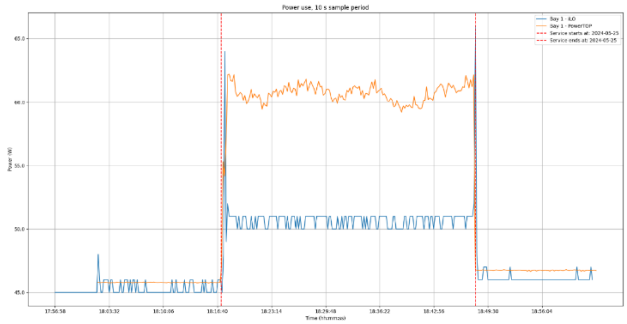

Figure 9. Twenty instances, containerized operations, baseline 1.

TABLE VI. MEAN POWER USE – FORTY SERVICE INSTANCES

Power type	Description	Avg. (W)
$\overline{p^{(iLO)}}[12:57:37,13:17:40]$	Before starting the service instance	45.56
$\overline{p^{(iLO)}}[13:17:40,13:49:15]$	During the service instance's operation	53.40
$\overline{p_{dyn}^{(ptop)}}[13:02:42,13:17:40]$	Mean dynamic power before service instances' operation	0.7206
$\overline{p_{dyn}^{(ptop)}}[13:17:40,13:49:15]$	Mean dynamic power during the service instances' operation	19.1873

E. Orchestation of native streaming

A similar set of experiments was run for native video servers. The results are available in the online repository, and are structured in the same manner as that used in Section V-D.

TABLE VII. MEAN POWER USE – EIGHTY SERVICE INSTANCES

Power type	Description	Avg. (W)
$\overline{p^{(iLO)}}[18:19:21,18:39:20]$	Before starting the service instance	45.53
$\overline{p^{(iLO)}}[18:39:30,19:13:53]$	During the service instance's operation	56.30
$\overline{p_{dyn}^{(ptop)}}[18:24:31,18:39:30]$	Mean dynamic power before service instances' operation	0.7435
$\overline{p_{dyn}^{(ptop)}}[18:39:30,19:13:53]$	Mean dynamic power during the service instances' operation	15.5243

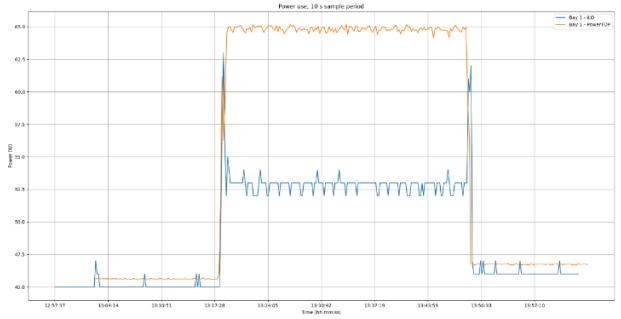

Figure 10. Forty instances, containerized operations, baseline 1.

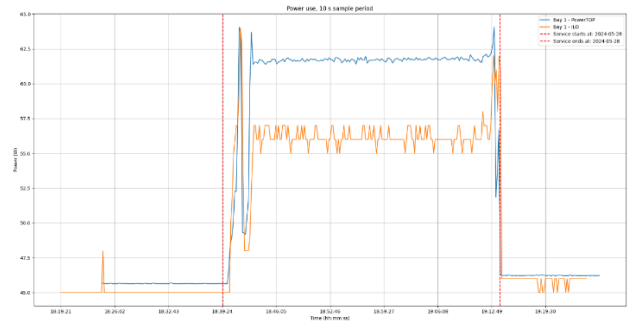

Figure 11. Eighty instances, containerized operations, baseline 1.

VI. ANALYSIS

Various characterizations of power use are considered and plotted in Figure 12. In the notation shown below, the (n) symbol indicates dependence of power used on number of streaming containers.

1. total power during operations, $P_{ops}^{iLO}(n)$, and
2. differential total power, where the difference is between operations and quiescence, $P_{ops}^{iLO}(n) - P_{q}^{iLO}$.

Figure 12 illustrates the results in graphical form. The top row of graphs compares total power and differential total power, respectively, for containerized and native operations. The bottom row shows the difference between total power and differential total power. The non-monotonic behaviour seen in the bottom row is due to the error introduced by the rounding of iLO instrumentation.

Dynamic power measurements as a function of streaming videos are not shown in Figure 12, as PowerTOP's measurements do not produce consistent, intelligible results on our platform. Estimates are insufficiently accurate. PowerTOP is capable of capturing power change behaviour (see, notably, Figure 11), but it requires further development before its estimates can be used for quantitative analysis.

VII. CONCLUSIONS

The objective set out in Section II was to quantify the overhead incurred by operating the video service containerized, instead of as an application running directly on the host operating system (native operation). An access network of the Active Ethernet type was constructed and a video cache deployed in an access node to stream videos to the access node's service area. An implementation model describing the access network was included.

The results obtained have shown that the overhead is negligible and that the benefit of running the video source in a container comes at little cost. The possibility of consolidating video streaming containers can be pursued with confidence.

No discernable cause for concern was found in the power measurement instrumentation embedded in the HPE Gen9 platform. Documentation on interfacing with the Integrated Lights-Out (iLO) server management was readily available. For detail beyond typical interest, HPE readily divulged information on this tool when contacted for help, including, for example, the method used to round the power measurement into an integer [23].

On the other hand, PowerTOP's accuracy poses a problem. The various graphs of power against time have shown that it captures changes well, but significantly overestimates them. In the light of these errors, works that have investigated containerization's overhead with the use of this tool (e.g., [24]) may need to be reviewed for the implications of inaccuracies introduced by the tool, perhaps by using external, physical power meters to calibrate PowerTOP's measurement.

Baselines have been obtained for both the video server and the virtual switch. In particular, P_{b2}^{video} has been found useful in obtaining a reproducible starting point for experiments; to a lesser extent, P_{bq}^{video} has been found useful in providing an offset for power obtained through tools that measure dynamic power. This segues well into an observation that merits particular attention. Even with 80 concurrent streams, the static power has dwarfed the dynamic power. The importance of this observation pertains to the importance of the benefit of containerization as an enabler of consolidation of physical hosts. It can readily be stated that the overhead incurred in providing the **service framework** of containerization poses no obstacle to exploration of exploitation of this benefit.

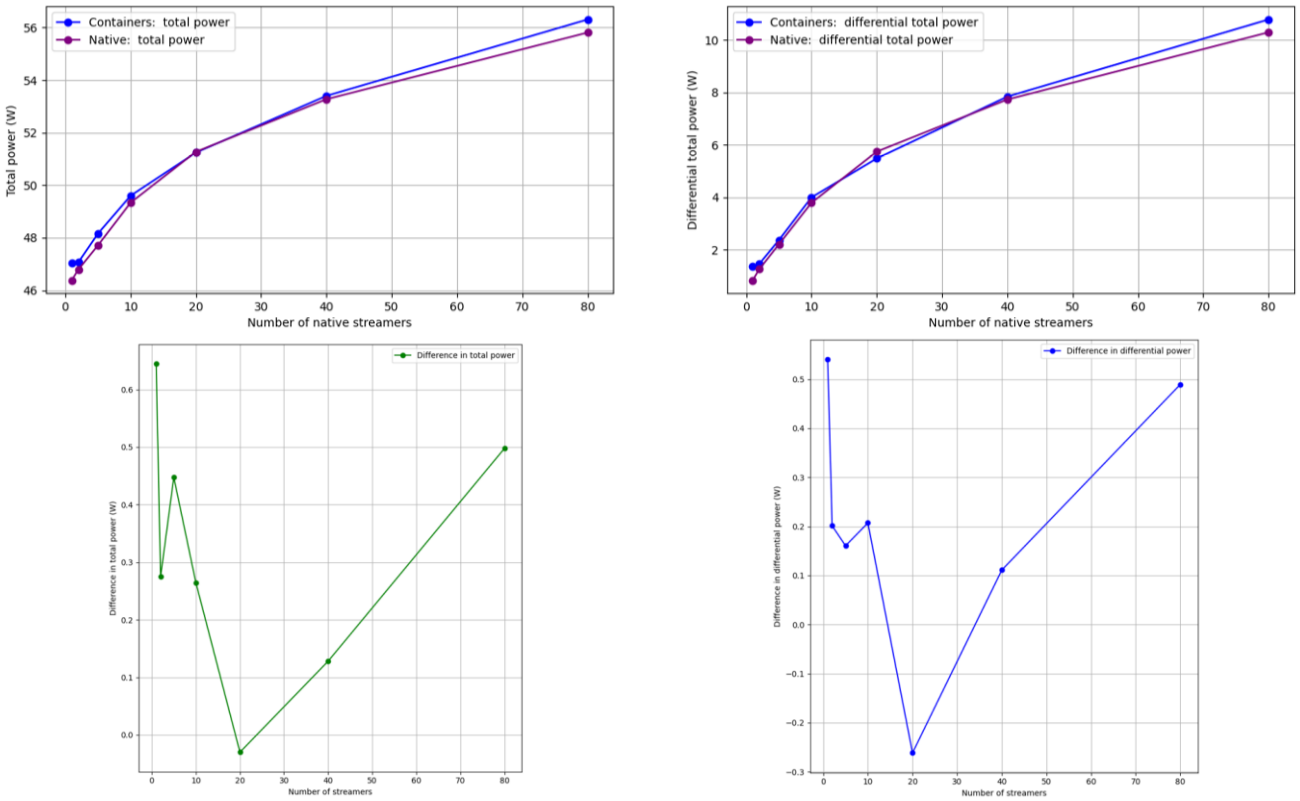

Figure 12. Comparison: native vs containerized streaming. Clockwise from top left: $P_{ops}^{iLO}(n)$, $P_{ops}^{iLO}(n) - P_{q}^{iLO}$, $P_{ops}^{iLO}(n_{cont}) - P_{ops}^{iLO}(n_{native})$ and $\left(P_{ops}^{iLO}(n_{cont}) - P_{q}^{iLO}{}_{cont}\right) - \left(P_{ops}^{iLO}(n_{native}) - P_{q}^{iLO}{}_{native}\right)$.

REFERENCES

[1] A. Teker, A. H. Örnek, and B. Canberk, "Network Bandwidth Usage Forecast in Content Delivery Networks", in *2020 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*, Jul. 2020, pp. 1–6. doi: 10.1109/CoBCom49975.2020.9174180.

[2] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers", *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013, doi: 10.1109/TIT.2013.2281606.

[3] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems", *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014, doi: 10.1109/MCOM.2014.6736753.

[4] S. Wang *et al.*, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications", *IEEE Access*, vol. 5, pp. 6757–6779, 2017, doi: 10.1109/ACCESS.2017.2685434.

[5] C. Mouradian *et al.*, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges", *IEEE Commun. Surv. Tutor.*, vol. 20, no. 1, pp. 416–464, 2018, doi: 10.1109/COMST.2017.2771153.

[6] J. Zhao, P. Liang, W. Liufu, and Z. Fan, "Recent Developments in Content Delivery Network: A Survey", in *Parallel Architectures, Algorithms and Programming*, H. Shen and Y. Sang, Eds., Singapore: Springer, 2020, pp. 98–106. doi: 10.1007/978-981-15-2767-8_9.

[7] B. Zolfaghari *et al.*, "Content Delivery Networks: State of the Art, Trends, and Future Roadmap", *ACM Comput. Surv.*, vol. 53, no. 2, p. 34:1-34:34, Apr. 2020, doi: 10.1145/3380613.

[8] "Multi-service Broadband Network Architecture and Nodal Requirements". TR-178 Issue 2, Broadband Forum, Sep. 2017. Accessed: Oct. 4, 2024. [Online]. Available: https://www.broadband-forum.org/technical/download/TR-178_Issue-2.pdf

[9] Hewlett Packard Enterprise, "HPE ProLiant BL460c Gen9 Server Blade", PSNow. Accessed: Jun. 17, 2024. [Online]. Available: https://www.hpe.com/psnow/doc/c04347343

[10] "FFmpeg". Accessed: Jun. 17, 2024. [Online]. Available: https://ffmpeg.org/

[11] "TSDuck". Accessed: Jun. 17, 2024. [Online]. Available: https://tsduck.io/

[12] "Open vSwitch". Accessed: Jun. 17, 2024. [Online]. Available: https://www.openvswitch.org/

[13] "index | Alpine Linux". Accessed: Jun. 17, 2024. [Online]. Available: https://alpinelinux.org/

[14] "Docker: Accelerated Container Application Development". Accessed: Jun. 17, 2024. [Online]. Available: https://www.docker.com/

 ISBN: 978-1-68558-203-6

[15] “Kubernetes Documentation”, Accessed: Sep. 03, 2019. [Online]. Available: https://kubernetes.io/docs/home/
[16] DMTF Redfish Forum, “Redfish Specification”. Apr. 03, 2024. Accessed: Jun. 17, 2024. [Online]. Available: https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.20.1.pdf
[17] “Powertop – ArchWiki”. Accessed: Feb. 20, 2024. [Online]. Available: https://wiki.archlinux.org/title/powertop
[18] A. van de Ven, private communication, “PowerTOP running average interval and display update interval”, Apr. 08, 2024.
[19] A. van de Ven, private communication, “PowerTOP running average interval and display update interval (update)”, Apr. 08, 2024.
[20] Ansible Community Documentation, “User Guide”. Accessed: Jun. 17, 2024. [Online]. Available: https://docs.ansible.com/ansible/latest/user_guide/index.html
[21] A. van de Ven, private communication, “PowerTOP running average interval and display update interval (2nd update)”, Apr. 30, 2024.
[22] E.V. Depasquale, “Video Streaming Power Use,” *GitHub Repository,* Oct. 26, 2024 [Online]. Available: https://github.com/edepa/video_streaming_power_use.
[23] J. Sultana, private communication, “Power measurement - Gen9”, May 24, 2024.
[24] R. Bolla, R. Bruschi, F. Davoli, C. Lombardo, and N. S. Martinelli, “Analyzing the Power Consumption in Cloud-Native 5/6G Ecosystems”, in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2023, pp. 611–617. doi: 10.1109/ICCWorkshops57953.2023.10283755.

Evaluating Trade-offs for Green Routing in Communication Networks

Jan Kitanovski*, Kaspar Zimmermann*, Line M. P. Larsen† and Sarah Ruepp*
* Department of Electrical and Photonics Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark
† Department of Mobile Innovation, TDC NET, Copenhagen, Denmark
e-mail: s232649@dtu.dk, s232759@dtu.dk, lil@tdcnet.dk, srru@dtu.dk

***Abstract*—With the demand for high-speed, high-capacity networking increasing each year, it is important to focus on the infrastructure and software running on the networking backbone. A crucial component of this focus is the need for efficient pathfinding algorithms, which can determine the best route for data to travel across a network, and ensuring optimal resource usage and performance. This paper promotes the importance of selecting the best routing algorithm for a specific purpose of data transport, highlighting the trade-offs involved in green routing. It also presents the work and results of a network simulator that uses four different algorithms (Dijkstra's, A*, Floyd-Warshall, and Depth-First Search) to determine which performs best in a set environment. In terms of latency, Floyd-Warshall showed a 64% improvement over Dijkstras, whereas A* showed a 57% improvement over Dijkstra's. Results indicate that the tradeoff of choosing an algorithm with a lower latency can also result in higher carbon cost.**

***Keywords*—Green-Routing; Denmark; Path-Finding; algorithms; latency; overhead; carbon cost.**

I. Introduction

As the number of Internet-connected devices continues to rise globally, the data throughput generated by these devices has significantly increased. In 2022, it reached nearly 1200 Tbit/s globally [1]. Consequently, new network infrastructure needs to be built, which also consumes energy. Data centers and other similar technologies used 460TWh in 2022, which accounted for almost 2% globally [2]. Therefore, it is important to focus on enabling greener alternatives to how the networks operate, as this might also impact on the end-cost of operating the infrastructure. Choosing green routing-themed research comes from the need to address the impact of the growing network infrastructure. With a big shift towards renewable energy sources, it is important to also optimise the network performance itself.

Routing is a way of finding the path from one node or end device, through the network to the end-point of the data. Green-routing improves on this, by also taking into account where the energy, powering the devices that enable this transmission comes from, and how it can be used in the most efficient way.

In this project, an event-based simulator will be programmed to simulate the network traffic within Denmark. The data used for the simulator will be based on the data centers and other smaller nodes spread throughout the country. Due to data gathering limitations, different alternative approaches had to be used to simulate the network behaviour.

*The first two authors have contributed equally to this paper

This paper describes a simulation project that focuses on green routing algorithms and compares them in terms of performance and efficiency.

The simulator operates as an event-based simulation, designed to manage large-scale simulations. Each event is assigned specific attributes before being queued for simulation. These attributes include the time of occurrence, which determines when the event will be processed, and the type of event. Event types include original packet creation, which initiates the start of the algorithm search. Normal packet creation can generate either an overhead packet or a data packet, depending on the algorithm's progress. Data packets can be generated with either high or low priority. High-priority packets are routed through the shortest path to minimize latency, while low-priority packets are directed through the greenest route to minimize environmental impact.

For the remainder of this paper, Section II provides the state of art, Section III dives into the methodology used in the event-based simulator, Section IV provides the Testbed setup of the simulation, Section V overlooks the results gathered from the simulation, which are then discussed in Section VI and concluded in Section VII.

II. State of the Art

Several studies have explored various aspects of the green routing problem, each approaching it differently. Zhu et al. [3] examined the development of an energy-aware network management platform, OpenNaaS, which supports SDN (Software Defined Networking) to create green-greedy routing paths. Their system measures energy, cost, and sustainability information for networks, demonstrating the platform's potential for energy-efficient routing in large-scale networks. Wang et al. [4] focused on power saving and QoS (Quality of Service) for many-to-many multicast in backbone networks. They developed the GIQM (Green Intelligent flexible QoS many-to-many Multicast routing algorithm), which uses power consumption as a routing metric and supports flexible QoS requirements. Their algorithm outperformed other schemes such as the CBT (Core-Based Trees algorithm) in power savings and routing success. In their research, Yang et al. [5] devised hop-by-hop algorithms to achieve loop-free routing, minimizing energy usage. They were innovated upon the Dijkstra's algorithm, by creating various "Dijkstra-green" versions to improve the routing efficiency. Lee et al. [6] proposed the DEAR algorithm, which improves energy efficiency while meeting flow delay requirements in networks with diverse energy profiles.

The DEAR algorithm effectively identifies the least energy-consuming path while ensuring flow delay requirements are satisfied. Hossain et al. [7] proposed a sustainable method for greening the Internet by introducing "pollution-aware routing," which integrates considerations of carbon emissions and non-renewable energy usage into traditional energy-aware routing. Their holistic approach, implemented with SDN, demonstrated significant reductions in CO2 emissions compared to conventional energy-aware solutions. Nwachukwu et al. [8] has focused on the optimization for simultaneous routing and bandwidth allocation through the use of Lagrangian methods. Their work shows that augmented Lagrangian algorithms are highly effective on this matter.

Upon reviewing existing research in this field, it was found that the primary focus has been on developing new technologies, algorithms, and enhancements to these algorithms, with relatively few studies dedicated to comparing the various path-finding algorithms.

The objective of this research paper is to develop an event-based simulator capable of incorporating various data, such as the locations of data center nodes within Denmark, and simulating network behaviour using different algorithms. The aim is then to compare the effectiveness of each algorithm based on multiple result variables, including total overhead packets, carbon cost per data transmitted, carbon cost distribution between data and overhead and latency.

III. Methodology

A. Green Routing

Green routing as a principle focuses on the ability of the network to choose transportation routes that minimize the environmental impact, by reducing fuel consumption and emissions. It utilizes different path-finding/optimization algorithms as well as data of the types of power sources available for powering the network. In this paper, the evaluation of these algorithms will be based on multiple values, such as the latencies, total overhead packets, total data packets, carbon emission of the transmission, expressed as the carbon cost. These values of course will be expressed as multiples of minimum, maximum and average values. These values will be represented in the corresponding figures.

B. Algorithms

In this research, the main focus will be on the comparison of results, efficiencies, and drawbacks of various path-finding algorithms. The algorithms in question will be: Dijkstra's, Floyd-Warshall, an advanced version of Dijkstra's, A*, and the DFS (Depth-First Search) approach. Figure 1 shows an example graph that the operation of the algorithms will be explained on.

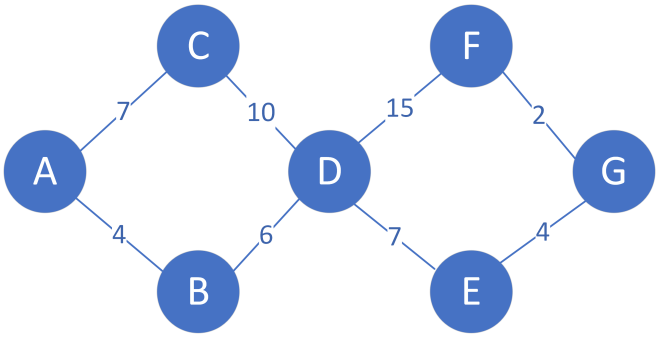

Figure 1. An example graph with 7 nodes and positive edges, used to illustrate algorithm operation

- Dijkstra's finds the shortest path starting from a single point and continuing by the use of the smallest known distance to other nodes in a weighted graph. When a shorter route to a node is found, the table is updated, until all paths to the end node are found. Starting from node A, the algorithm will choose the shortest distance, which is to node B. The next shortest step is to node D, after which the algorithm sets the shortest distance to D as 10. The algorithm keeps taking the shortest path available to it and when it gets to a node that is already discovered, it will update the distance to it, if it is shorter. For example, when the path from C to D is discovered, its distance of 17 is compared to the previously found distance of 10, and no changes are made [9]. Dijkstra's is tested because it is a very commonly used shortest-path algorithm and is easy to implement.
- A* extends Dijkstra's algorithm by incorporating heuristics, which are used to guide the search towards the goal, and utilizes tables to keep track of visited and non-visited nodes. When A* finds the path in the Dijkstra's searching method, it will save it. For example, it will know that the path from A to D is shortest through B, so next time it won't search the entire graph again and will use that path [10]. A* is used to see if it is possible to fix some of the shortcomings of Dijkstra's, but not lose the accuracy.
- The Floyd-Warshall algorithm calculates the shortest paths between all pairs of nodes in a weighted graph. It initializes the distance matrix with the weights of direct connections between nodes, using infinity for pairs of nodes without a direct connection and zero for the distance from a node to itself. The algorithm then updates this matrix by considering each node as an intermediate point and checking if a path through this node offers a shorter route between any two nodes. For example in the given graph, the algorithm starts by setting the initial distances, such as A to C to 7, A to B to 4, and A to D to infinity. It then updates paths A to D by taking into account intermediate nodes. For example, it might find that the path from A to D through C (with a combined weight of 17) or through B (with a combined weight of 10) is shorter than the initially set distance. In this case, the path A to B to D has a total weight of 10, which would be the updated shortest distance from A to D [11]. Floyd-Warshalls algorithm was chosen, because it is a competing shortest-path algorithm to Dijkstra's. Their difference should mostly depend on the density of

the graph, so results may change depending on the chosen network size.

- DFS initiates by choosing a point of entry and randomly continuing node by node in one direction. If a connection is not successfully established, it backtracks its hops and tries again. Starting from node A, it will randomly choose node C or B. Then, it will keep going with random choices, without backtracking. If it finds the end node, it will use whatever path it took even if its not the most efficient one. DFS was chosen as a reference point to use for comparing the other algorithms. It shows why, in general, using a routing algorithm is better than randomly choosing which way to go.

IV. Testbed Setup

Python was used to set up the simulation. Additional libraries, such as heapq and pandas were used for efficient priority queue operations and data manipulation and analysis, respectively, while Matplotlib was used to visualise the results.

The data for the simulator was gathered from various sources, such as different research papers and publicly available data, which are explained below. However, very precise data was not available, so reliance on other models for electrical energy sourcing, including the estimation of the ”greeness of sources,” and the connection between the bandwidth created and population densities in certain areas was necessary.

The carbon cost calculation was done with the use of the nearest energy sources to the node and multiplied by the total distance travelled. The data has been gathered by the Danish energy agency [12].

The latency calculation was done by combining the data for the propagation of light through optic fiber of 500 microseconds per 100km [13] and the average latency introduced by the processing time of a switch set to 2 microseconds [14].

It is worth noting that the connections between datacenters in Denmark were established based on a publicly available map [15], which was transcribed into the setup shown in Figure 2. For the simulator itself, a few different configurations were chosen, however the main difference between them was the number of simulations to run.

The estimation for electrical energy sourcing was derived from a model using the closest sources of electrical power. In this approach, a specific area around the data center, with an estimated set power consumption, was examined. A percentage-wise distribution of power sources, which could be from renewables or non-renewables, was utilized. A similar approach was used for estimating the amount of data generated at a certain node in the network. This estimation was based on the map of Denmark, with population densities being utilized for the calculations.

The tests are run for 10,000 seconds to minimize the effect of the semi-random packet generation without losing the realistic traffic. All tests were run at least 5 times to check for anomalies. This was normally enough, as the long simulation time removed any randomness from the results.

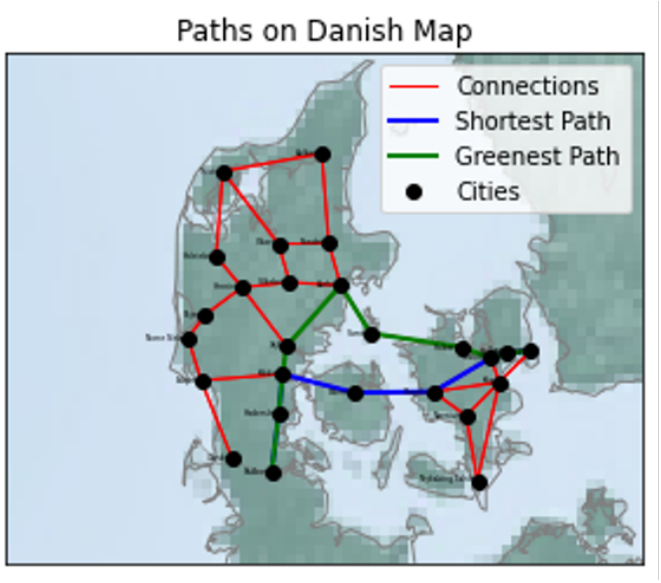

Figure 2. The main network nodes of Denmark, showing the use of Dijkstra's algorithm for path-finding the lowest latency route, as well as the greenest path.

The graph used for the algorithm uses the map of databases and also places nodes in locations where fibre optic cables meet at larger settlements. It consists of 26 points with each having 1 to 5 connections to nearby nodes.

V. Results

For the results, three main metrics were considered: Overhead packets, which show non-data packets used by each algorithm, Carbon cost, to show how ”Green” the algorithm is and Latency to show how fast the algorithm is.

1) Overhead packets: The test results for total overhead packets created by the algorithms in the simulation time are shown in Figure 3. The amount of overhead packets created can indicate how much additional traffic the algorithm creates for the entire network.

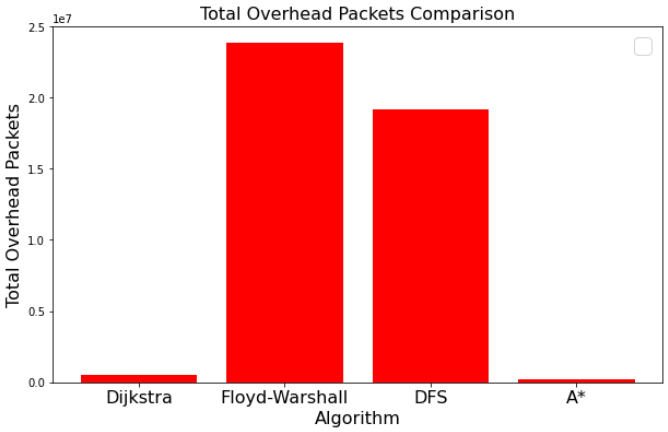

Figure 3. Total overhead packets sent by each algorithm during the duration of the simulation

Floyd-Warshall produces the highest amount of overhead because it uses overhead packets to find the cost between every pair of nodes. The random design of the DFS approach creates around 23% less overhead than Floyd-Warshall. Both of these approaches created significantly more overhead than the Dijkstra-based approaches and would require a lot more processing by all nodes in the network.

Dijkstra's algorithm creates 97.86% less overhead than Floyd-Warshall and therefore causes less congestion of the network. The implementation of A* creates 53% less overhead than Dijkstra's and 99% less overhead than Floyd-Warsall, which is expected, as even though it searches all connections in the network, it only does it every 250 seconds. Using the saved routes later means that no extra overhead is created and the capacity can instead be used for data. This might cause problems if a larger part of the network is disconnected right after the full search, which is why the precise time of searches should be balanced according to the network requirements.

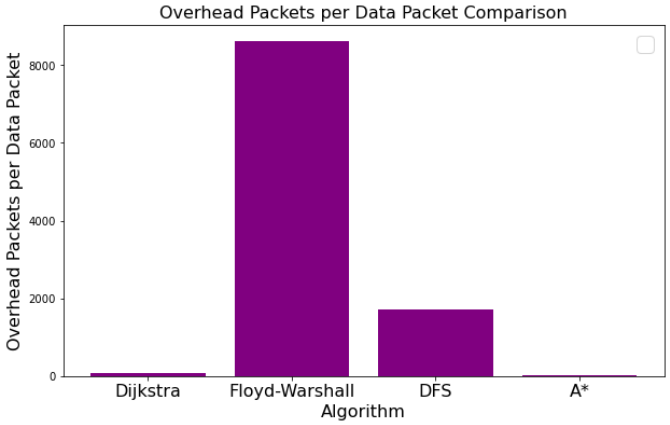

Figure 4. Total Carbon cost per data packet sent by each algorithm during the simulation

Figure 4 illustrates that DFS sends 80% fewer overhead packets per data than Floyd-Warshall. This is due to the non-optimal paths that the data chooses in the DFS algorithm, which means the data travels more in the graph.

2) Carbon costs: The results from the carbon cost calculations can be seen in Figure 5. As overhead packets carry less information than data packets, the carbon cost of overhead is set to 1/1000 of data.

The largest carbon cost is produced by the DFS algorithm. It uses a lot of carbon for overhead packets used for searching, and as the routes it finds are not the most optimal, it also uses the most on data. Although Floyd-Warshall uses the least amount of carbon on data, its large amount of searching means that it has the second highest carbon cost, which is still 99% less than DFS. It is evident though that if the overhead packets were even smaller compared to data, then Floyd-Warshall could have the smallest carbon cost. In this simulation, A* generates 40% less overhead than Floyd-Warshall. Since the graph of nodes is quite small, the performance of A* is worse than Dijkstra's by 38% as the benefits of the heuristic model

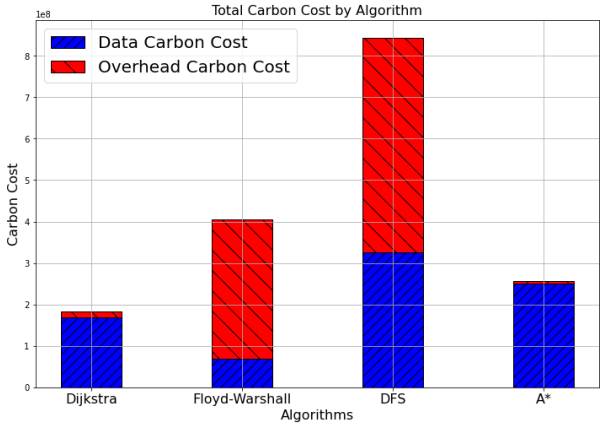

Figure 5. Carbon cost distribution showing each algorithms carbon cost from data and overhead packets

do not appear. The carbon cost of Dijkstra's is the smallest, although the paths it finds are not the most optimal, it uses less overhead than Floyd-Warhsall, which makes it overall 57% more "Green".

3) Latency: The third parameter that shows the efficiency of the algorithm is latency. Latency in this case takes into account the search time and the data travel time. Simulation results are shown with maximum and minimum limits in Figure 6.

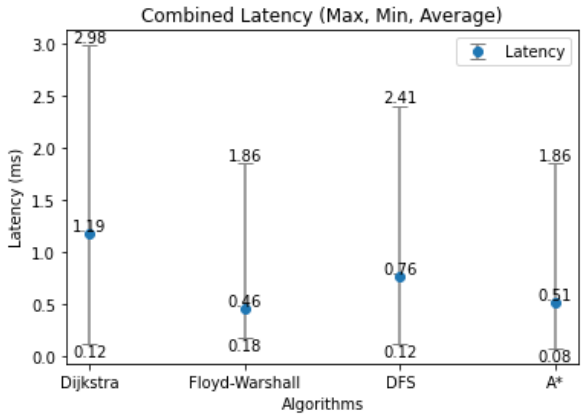

Figure 6. Latency results with average latencies and maximum and minimum latencies shown as limits

The largest average and maximum latency is by Dijkstra's algorithm, which is 55% more than the second-best DFS. The search starts from one point and systematically searches through the entire network. Floyd-Warshall has the lowest average latency due to how the algorithm searches, with a 64% improvement on Dijkstra's. As it starts from several nodes at the same time, it is faster than Dijkstra's. However, as it still needs to search most of the graph, it has the highest minimum latency. DFS has the second-highest average and maximum latency. This is due to the randomness, which worst case will search the entire graph, but on average will only search half of it. A* has the lowest minimum latency due to using pre-

saved paths, which do not require searching. The case where A* searches the entire network increases the average latency, but it is still on average 57% faster than Dijkstra's.

VI. Discussion

These results highlight that with each algorithm there are positives and negatives. If the impact of overhead packets is negligible and their size is small, then only the efficiency of the routing is important. In this case, Floyd-Warshall could be used for extensive searching. In case the goal is to minimize the amount of overhead, using a version of A* or Dijkstra's is preferable. However, when choosing Dijkstra's the network will have longer latency, which should not be used for real-time communication. A* can solve that problem with the use of saved paths and heuristics. To make A* more efficient, a larger network is needed, where the model can improve on Dijkstra's. The tradeoffs of each algorithm are illustrated in Figure 7. With the X-axis showing increasing total carbon emissions and the Y-axis showing the average latency, it is best if the algorithms aren't in the extremes in any axis. Dijkstra's is on the top left of the graph, which means that although the total carbon emissions are low, the latency is significantly worse than the others. Similarly, DFS is on the extreme of carbon emissions. Floyd-Warshall and A* both do rather well, but there is some noticeable difference in carbon emissions. In general, as A* has a suitable average latency and low carbon cost, it can be preferred over the other algorithms tested.

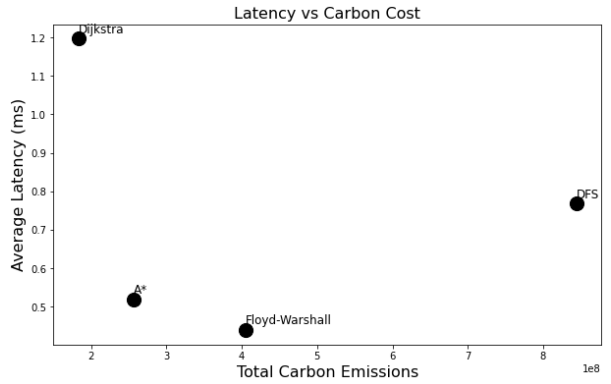

Figure 7. Carbon cost and latency comparison for all algorithms with carbon cost on the x-axis and latencies on the y-axis

VII. Conclusion

This paper was set to address the environmental impact of the expanding ICT (Information and Communication Technology) infrastructure by comparing different path-finding algorithms.

The findings indicate that the Floyd-Warshall algorithm produces the most overhead packets, followed by Depth-First Search, with a decrease of 23%, while A* has the fewest, with 99% less overhead packets than Floyd-Warshall. In terms of carbon cost, the DFS algorithm generates the highest total carbon cost, and A* having a 38% worse performance than Dijkstra's. Interestingly, Floyd-Warshall has a lower carbon cost for data packets than Dijkstra's; however, the higher number of overhead packets increases its total carbon cost. Regarding latency, Floyd-Warshall has the lowest average latency with an improvement of 64% over Dijkstra's, while Dijkstra's shows the largest range of values. A* also shows a 57% improvement over Dijkstra's, even in the situation, where it searches the whole network.

These results demonstrate that no single algorithm is universally optimal, as each one has its strengths and weaknesses. However, as A* has good latency and carbon cost metrics, it can be the preferred algorithm for most cases.

Acknowledgment

Support from Innovation fund Denmark, through grant no. 1045-00047B, as well as the "Innovative solutions for next generation of Green COMmunications infrastructures" GREENCOM project is gratefully acknowledged.

References

[1] ITU, "Unrelenting global consumption of Internet data continues to drive demand for international bandwidth usage," https://www.itu.int/itu-d/reports/statistics/2022/11/24/ff22-international-bandwidth-usage/ [retrieved: June, 2024], 2022.

[2] IEA, "Electricity - Analysis and forecast to 2026," https://iea.blob.core.windows.net/assets/6b2fd954-2017-408e-bf08-952fdd62118a/Electricity2024-Analysisandforecastto2026.pdf[retrieved: June, 2024], 2024.

[3] H. Zhu, J. Aznar, C. de Laat, and P. Grosso, "Green routing in software-defined data center networks based on opennaas," *Software Networks*, 2015.

[4] X. Wang, J. Zhang, M. Huang, and S. Yang, "A green intelligent routing algorithm supporting flexible qos for many-to-many multicast," *Computer Networks*, vol. 126, pp. 229–245, 2017.

[5] Y. Yang, M. Xu, D. Wang, and S. Li, "A hop-by-hop routing mechanism for green internet," *Ieee Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 2–16, 2016.

[6] E. J. Lee, Y. M. Kim, and H. S. Park, "Dear: Delay-guaranteed energy profile-aware routing toward the green internet," *Ieee Communications Letters*, vol. 18, no. 11, pp. 1943–1946, 2014.

[7] M. M. Hossain, J. P. Georges, E. Rondeau, and T. Divoux, "Energy, carbon and renewable energy: Candidate metrics for green-aware routing?" *Sensors (switzerland)*, vol. 19, no. 13, p. 2901, 2019.

[8] A. C. Nwachukwu and A. Karbowski, "Solution of the simultaneous routing and bandwidth allocation problem in energy-aware networks using augmented lagrangian-based algorithms and decomposition," *Energies*, vol. 17, no. 5, p. 1233, 2024.

[9] GeeksforGeeks, Sanchhaya Education Private Limited, "What is dijkstra's algorithm? — introduction to dijkstra's shortest path algorithm," https://www.geeksforgeeks.org/introduction-to-dijkstras-shortest-path-algorithm/ [retrieved: June, 2024], 2024.

[10] ——, "A* search algorithm," https://www.geeksforgeeks.org/a-search-algorithm/ [retrieved: June, 2024], 2024.

[11] ——, "Floyd warshall algorithm," https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/floyd-warshall-algorithm [retrieved: June, 2024], 2024.

[12] The Danish Energy Agency, "Power production and transmission in denmark," https://ens.dk/en/our-services/statistics-data-key-figures-and-energy-maps/energy-infomaps [retrieved: June, 2024], 2019.

[13] F. Azendorf, A. Dochhan, and M. H. Eiselt, "Accurate single-ended measurement of propagation delay in fiber using correlation optical time domain reflectometry," *Journal of Lightwave Technology*, vol. 39, no. 18, pp. 5744–5752, 2021.

[14] T. Hegr, M. Voznak, M. Kozak, and L. Bohac, "Measurement of switching latency in high data rate ethernet networks," *Elektronika Ir Elektrotechnika*, vol. 21, no. 3, pp. 73–78, 2015.

[15] DataCenterJournal, "Map of Denmark's Data Centers," https://www.datacenterjournal.com/data-centers/denmark/ [retrieved: June, 2024],, 2024.

Measurement Based Time-Domain Power Saving Through Radio Equipment Deactivation on Sub-6GHz Base Station Site

Youssef Agram*, François Rottenberg†, and François Quitin*
*Brussels School of Engineering, ULB, Brussels, Belgium
†Faculty of Engineering Technology, KULeuven, Ghent, Belgium
e-mail: *{youssef.agram, francois.quitin}@ulb.be,
†{francois.rottenberg}@kuleuven.be

***Abstract*—The paper presents a methodology for reducing the power consumption of radio access networks by scheduling the use of radio equipment from multiple radio access technologies while keeping comparable quality of service. Although 5G New Radio (NR) is known to be more energy-efficient than legacy 4G Long Term Evolution (LTE) equipment, previous work shows a higher static power for 5G-NR equipment using active antennas. In the context of increasing data traffic and the deployment of 5G-NR equipment along 4G-LTE, the question of concurrent energy-efficient use of these two technologies for a given traffic load arises. This analysis relies on on-site measurements from a macro sub-6GHz base station in Belgium, evaluating the energy efficiency of 4G-LTE and 5G-NR radio units. Our findings demonstrate that the 5G-NR equipment is more energy-efficient at higher traffic levels, i.e., $\geq$ 150 Mbps, while using 4G-LTE is preferable at lower traffic due to its smaller static power consumption. Besides, the dynamic energy-efficiency is 3 to 9 times higher for 5G-NR compared to 4G-LTE. The paper also proposes several radio unit deactivation scenarios: 1) using 4G-LTE radio units only and redirecting traffic on 5G-NR when reaching 80% of the maximum 4G-LTE capacity, 2) using 5G-NR radio units exclusively, and 3) dynamically selecting between 4G-LTE and 5G-NR based on the computed downlink data traffic threshold. The results show that Scenario 3 achieves the largest energy savings, reducing power consumption by 31.5% at the base station level. This paper demonstrates that it is possible to significantly reduce the energy footprint with equipment that are currently deployed.**

***Keywords*-power consumption; base station; measurements; deactivation; radio units.**

I. Introduction

The Radio Access Network (RAN) plays a critical role in mobile communications, including a large number of Base Stations (BSs) which are responsible for over 80% of the total energy consumption in mobile networks [1][2]. In addition, historical trends indicate that RAN energy consumption is still increasing, requiring concrete action from mobile operators who are constrained to reduce their carbon footprint and ensure compliance with climate targets [3][4][5]. Besides, 5G New Radio (5G-NR) infrastructures are being widely deployed, offering enhanced features designed to improve both quality of service and energy efficiency. These features include deeper sleep modes, leaner carrier design, higher bandwidth, etc., when compared to their legacy counterparts [6][7]. Whereas multiple Radio Access Technologies (RATs) add extra layers of complexity to power consumption handling, the concurrent operation of 4G Long Term Evolution (4G-LTE) and 5G-NR equipment within existing RANs also presents a new set of challenges and degrees of freedom for energy savings [7]. While 5G technology is recognized for its superior energy efficiency at high traffic loads, it also exhibits a higher static power consumption compared to 4G-LTE at lower traffic levels [8]. This complementarity necessitates a strategic approach to schedule the dual-use of 4G-LTE and 5G-NR equipment from an energy-aware perspective, leveraging the strengths of each technology, without compromising the quality of service provided to end-users. Meanwhile, time-domain power saving features seem to provide the highest power saving gains [6].

Figure 1 shows on-site measured average Radio Frequency (RF) power consumption as a function of the average physical resource load for 2 Radio Units (RU) running different protocols, i.e., 4G-LTE for the Remote Radio Unit (RRU) and 5G-NR for the Active Antenna Unit (AAU). The intersection between the linear extrapolated trends indicates that it would be more energy-efficient to use one radio equipment rather than the other depending on the physical load, i.e., the 4G-LTE RRU at low and 5G-LTE AAU at high loads. It is, however, important to emphasize that this threshold does not correspond to the same service provided by both RUs. In fact, both have a different maximum capacity, e.g., bandwidth, number of layers, etc., and therefore, will not deliver the same downlink data traffic, even when the load is identical.

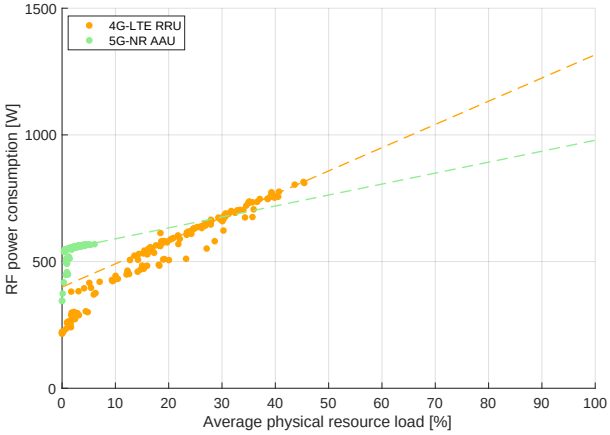

Figure 1. Average power consumption vs. average physical resource load measurements on an hourly basis. Inflexion of the curve is due to implemented power-saving modes [8].

This work aims to derive a simple energy-aware mechanism which deactivates specific RUs based on the downlink data rate, requested to the base station by User Equipments (UEs). This analysis assumes that UEs are both 4G-LTE and 5G-NR compatible. The purpose is also to compare energy-efficiency of equipment from both technologies and to quantify the energy savings achieved through the implementation of such mechanism.

This paper is organized as follows. Section II describes the architecture of the base station of interest, as well as the type and structure of available data. Section III recalls the BS power model from previous work, presenting a generic linear power consumption model for RUs, as a function of the physical resource load. Analysis and computation of the downlink data rate are performed in Section IV, before showing the relationship between power consumption and data traffic for all RUs. The implementation of the power saving mechanism through RU deactivation is illustrated in Section V, where several scenarios are proposed. Section VI concludes this work and lists future ones.

II. Structure of the base station

Measurements on which this study relies on are provided for an up-to-date 3-sector macro-BS deployed in a large city in Belgium. This site is equipped with 3 types of RUs:

- Radio Frequency Unit (RFU), installed in the BS cabinet and usually serving all 3 sectors at the same time,
- Remote Radio Unit (RRU), installed closer to passive antennas and dedicated to a specific sector,
- Active Antenna Unit (AAU), combining Analog Front-End (AFE), Power Amplifiers (PAs) and antenna elements in a single unit.

It supports three bands for LTE (i.e., 0.8, 1.8 and 2.1 GHz) and two bands for NR (i.e., 0.7 and 3.5 GHz). Figure 2 shows a simplified version of the BS architecture where each RU_i is enumerated with index $i \in \{1, 2, ..., 8\}$. The Digital Baseband (DBB) performs digital signal processing operations and provides data and control signals to RUs using the Common Public Radio Interface (CPRI) [9][10][11]. The considered BS comprises two separate System Modules (SM). Power Supply and Cooling systems (PSC) are also installed on site but are not represented.

On-site measurements have been performed by mobile operators on an hourly basis over one week (6 days) in 2023. Several metrics are covered such as the number of used Physical Resource Blocks (PRB), the number of UEs having reported a Channel Quality Indicator (CQI) with index X_j, etc. The majority of those measurements are given per *cell*, i.e., per frequency band and per sector. Only power consumption is aggregated at the RU level. For confidentiality reasons, the raw dataset cannot be published.

III. Power models

Previous work already developed a detailed parametric model for the BS power consumption, expressed as sum of the main BS components [8][10]:

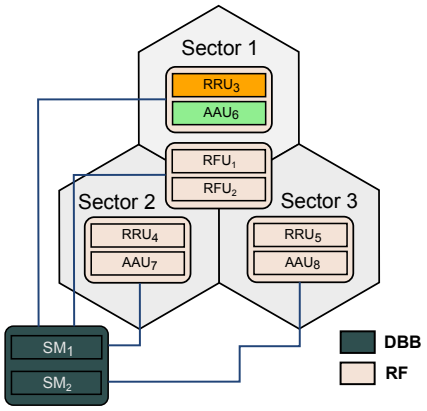

Figure 2. Simplified architecture of the base station of interest.

$$P_{BS} = \sum_{c=1}^{N_C} \left(P_{RU,c} + P_{DBB,c} + P_{PSC,c} \right), \tag{1}$$

where N_C denotes the number of cells and c the cell index (here and in the rest of this paper). It also demonstrates that RU equipment dominates in terms of power consumption. We will, therefore, be focusing on RUs in this study, which could themselves be split into the AFE and PAs. The average hourly RU power consumption can be expressed as:

$$\begin{aligned} \overline{P}_{RU}(T_k) &= \sum_{c=1}^{N_C} P_{PA}(\overline{x_c}(T_k); \chi_{PA,c}(T_k)) + P_{AFE}(\chi_{AFE,c}(T_k)) \\ &= \sum_{c=1}^{N_C} \alpha(\chi_{PA,c}(T_k)) \cdot \overline{x_c}(T_k) + \beta(\chi_{AFE,c}(T_k)) + P_{AFE}(\chi_{AFE,c}(T_k)), \end{aligned} \tag{2}$$

where T_k is the time sample corresponding to k^{th} hour on a given day. χ_{PA} and χ_{AFE} denote the set of configuration parameters for a given sub-component, e.g., number of active PAs, time ratio of downlink mode, etc., $\overline{x_c}$ is the average load for a given cell, α and β the load-dependent and static PA power consumption. Expression (2) is linear with respect to the load, in line with Figure 1 and [12].

The average physical resource load for a given cell is itself given by:

$$\overline{x_c}(T_k) = \frac{\overline{N}_{PRB,c}(T_k)}{\overline{N}^{tot}_{PRB,c}(T_k)}. \tag{3}$$

$\overline{N}_{PRB,c}$ (respectively $\overline{N}^{tot}_{PRB,c}$) denotes the average used (respectively total available) number of PRBs.

The issue with the above equations is that they do not explicitly involve the data rate, which is a metric that reflects user data requests and Quality of Service (QoS). This will be addressed in the following section.

IV. Data rate analysis

The purpose of this subsection is to derive a relationship between the RU consumed power and the downlink data rate.

A. Computation

Data traffic computation can be confusing, given the complexity of 4G-LTE and 5G-NR protocols. In this case, the targeted traffic is that at the input of the physical layer. Even so, the data rate can be calculated in different ways depending on the technology, e.g., Transport Block Size (TBS) table in 4G-LTE [13]. However, an estimation of the instantaneous data rate per cell can be derived from [14]:

$$R_c^{5G}(t) = k_c \cdot N_{L,c} \cdot Q_m \cdot r \cdot \frac{N_{PRB,c}^{\mu}(t) \cdot 12}{T_s^{\mu}} \cdot \left(1 - OH^{5G}\right), \quad (4)$$

where $N_{L,c}$ is the supported number of layers in the cell, Q_m denotes the modulation order in bits/symbol, r is the code rate, OH the overhead due to PHY signaling and $N_{PRB,c}^{\mu}$ is the number of used PRB in the cell within T_s^{μ}, the duration of an OFDM symbol with given numerology factor μ. k_c is a scaling factor to adapt for MIMO layers, which is assumed to be 1 here [14]. The ratio $\frac{N_{PRB}^{\mu} \cdot 12}{T_s^{\mu}}$ represents the symbol rate, with 12 being the number of subcarriers contained in a 5G-NR PRB. $Q_m \cdot r$ is also known as the efficiency [13][15].

The above equation cannot be used as such to get an average data rate on an hourly scale for several reasons:

- it assumes a single efficiency factor. Yet, it is evident that thousands of user requests are sent per hour, each with a different channel quality. An hourly CQI distribution should, therefore, be considered, with $\mathbb{P}_c^i(X_j, T_k)$ being the probability of having CQI X_j, within hour T_k, on RU_i and cell c. CQI indexes X_j range from 0 to 15, leading to $N_{CQI} = 16$,
- the definitions of 4G-LTE and 5G-NR PRBs are different, 84 (respectively 12) subcarriers are contained in an LTE (respectively NR) PRB,
- a cell can itself be shared by multiple mobile operators, leading to multiple *logical cells*, denoted by index c'.

Assuming that every RU runs a single technology, the average downlink data rate for a given RU_i can thus be expressed as:

$$\overline{R}^i(T_k) = \sum_{c'=1}^{N_{C'}^i} \sum_{j=1}^{N_{CQI}} \alpha_{c'}^i \cdot Q_{m,j}^i \cdot r_j^i \cdot \frac{\mathbb{P}_{c'}^i(X_j, T_k) \cdot N_{PRB,c'}^i(T_k) . N_s^i}{T_k}, \quad (5)$$

where $N_{C'}^i$ indicates the number of logical cells of RU_i and $\alpha_{c'}^i = N_{L,c'}^i \cdot (1 - OH^i)$. N_s^i denotes the number of subcarriers per PRB for a given technology running on RU_i. $Q_{m,j}^i$ and r_j^i are the modulation order and the code rate for a given CQI X_j and RU_i. Both depend on index i because 4G-LTE and 5G-NR RUs rely on different CQI standard tables, i.e., Table 7.2.3-1 (64QAM) for 4G-LTE [13] and Table 5.2.2.1-3 (256 QAM) for 5G-NR [15].

Note that one should, in theory, consider the Modulation and Coding Scheme (MCS) rather than the CQI to compute the efficiency. In fact, the scheduler could be using link-adaptation and lower the MCS to ensure an even lower Block Error Rate (BLER) [16][17]. Unfortunately, the chosen MCS counters are not given in the data source. Useful RU technical parameters are given in Table I. All RUs are single-band except 4G-LTE RRUs, which are dual-band.

B. Results

The initial analysis focus on examining the average CQI distribution per RU. For this purpose, the average Cumulative Density Function (CDF) is calculated over the week as such:

$$F(X) = \sum_{\substack{j=1 \\ X_j < X}}^{N} \frac{1}{N_K} \sum_{k=1}^{N_K} \mathbb{P}(X_j, T_k), \quad (6)$$

where $N_K = 144$, the number of samples over 6 days. The CDFs are given in Figure 3. Since 4G-LTE and 5G-NR RUs rely on different CQI tables, it is necessary to project CQIs on the same table to make curves comparable. This is achieved by converting CQI indexes of each table into Signal-to-Noise Ratio (SNR), using appropriate relationships [19]. The results show that users served by 5G-NR RUs, on average, send a higher CQI back to the BS, indicating a higher channel quality. Several explanations are conceivable for such results: these could be due to a scheduler selection bias which intentionally selects UEs with higher SNR for 5G-NR, or to closer location of 5G-NR with respect to the BS, or to UEs with more robustness against interference.

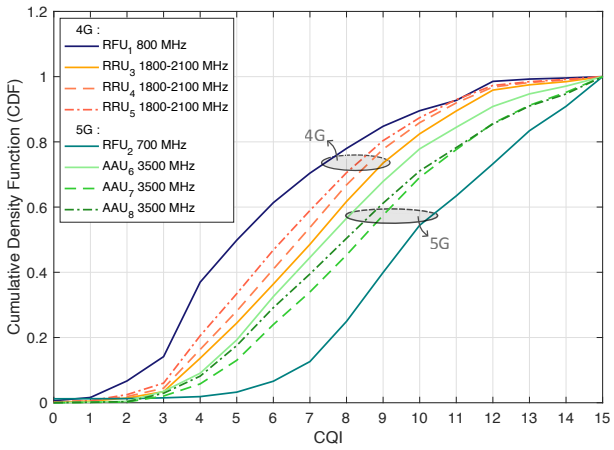

Figure 3. Average cumulative density function of reported CQI over the week (6 days), from Table 5.2.2.1-3 (256 QAM) [15].

Now, let us consider the relationship between the RU power consumption and the downlink data rate. Combining (2), (3) and (5) would lead to a linear relationship between $\overline{R}$ and $\overline{P}_{RU}$, since they all linearly scale with the number of used PRBs. This is in fact what is shown in Figure 4.

The slopes on this graph represent the dynamic energy intensity in J/Mb, which corresponds to the inverse of the energy efficiency in Mb/J. It mainly depends on the average channel quality and on the RU capacity, i.e., $C = N_L \cdot B$, with B representing the bandwidth. Although they have the same theoretical capacity, RFU_1 has a lower energy efficiency

TABLE I. RU TECHNICAL AND MODEL PARAMETERS.

RU type	Technology	f[1] [GHz]	$B_{C'}$[2] [MHz]	N_S[3]	$N_{C'}$	$N_{L,c'}$	N_s	OH	α [J/Mb]	P_{stat} [W]
RFU_1	LTE	0.8	10	3	6	2	84	0.11[4]	5.1	373
RFU_2	NR	0.7	10	3	6	2	12	0.14[5]	1.5	275
$RRU_{\{3,4,5\}}$	LTE	1.8 | 2.1	20	1	4	4|4	84	0.11[4]	1.8	345
$AAU_{\{6,7,8\}}$	NR	3.5	100	1	2	4	12	0.14[5]	0.2	548

[1] Carrier frequency [2] Bandwidth [3] Number of served sectors [4] [18] [5] [14]

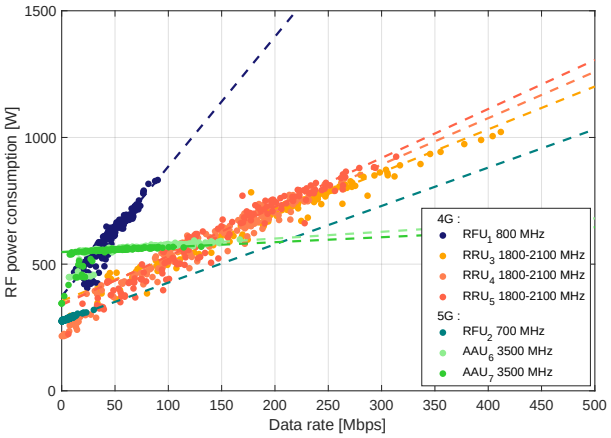

Figure 4. RU power consumption vs. data rate. Dashed lines represent linear regression models of scatter plots.

than RFU_2, due to its poorer average CQI distribution. Furthermore, AAUs have a very good energy efficiency, due to the combination of a higher capacity and a better channel quality. The absolute values indicate that AAUs should be preferred to RRUs at higher data rates regarding their lower power consumption, and vice versa. Note that AAU_8 is not shown due to a problem with power measurement that week. The dynamic energy intensities and static power values, denoted by α and P_{stat} are shown in Table I. It shows 3 to 9 times higher dynamic energy efficiency for 5G-RUs compared to its 4G-LTE counterparts.

In the next section, we examine how to leverage these results to implement our power reduction mechanism.

V. Power Savings through Deactivation

This section proposes a radio equipment time-domain deactivation methodology and provide gain margins at the base station level for different mechanisms.

A. Methodology

The previous results validated the need to develop a power reduction mechanism. Several type of power-saving features exist and can be implemented, such as [6][8]:

- time-domain technique: switching of components from working to idle or sleep mode,
- frequency-domain technique: deactivation of a half or full frequency band,
- spatial-domain technique: deactivation of half of the layers and of the corresponding PAs and TX/RX chains,
- power-domain technique: reducing PA transmission power and PA efficiency improvement.

Here, we only focus on time-domain power-saving technique on a larger time scale, i.e., hourly, by switching-off radio equipment corresponding to one technology at a time. The users connected to these RUs must therefore be redirected to other active devices. The followed methodology includes some constraints and assumptions:

1) UEs should be rerouted on a RU that belongs to the same sector as the previous one, to prevent deteriorating the channel quality,
2) UEs should remain in the same band types, i.e., coverage bands ({700, 800} MHz) or high-bands ({1800, 2100, 3500} MHz), to prevent deteriorating the channel quality,
3) UEs must preserve their SNR and thus their CQI when redirected,
4) UEs are assumed to be both 4G-LTE and 5G-NR compatible,
5) a deactivated RU still consumes some residual power due to part of the AFE used to reactivate it by the DBB [8].

To satisfy the first 2 constraints, let us denote $\mathcal{I}$ the set of all index pairs of RUs between which rerouting is feasible. Based on Figure 2, $\mathcal{I} = \{(1,2),(3,6),(4,7)\}$. Sector 3 is not considered in this section due to the lack of power measurements on AAU_8. From there, a decision threshold based on the total downlink hourly data rate is required. The linear regressions of Figure 4 cannot be used as such to find a threshold because the slope for each RU depends on the CQI distribution reported by the users connected to it, which we know differs between RUs. Projecting the data rate between pairs of RUs would change the average user's CQI, which would violate the 3rd constraint.

One solution is to build a global linear regression model, where each RU model also takes into account the CQI of the one with which it is paired. Such model can be expressed in matrix form:

$$\begin{bmatrix}\overline{P}_{RU_i}\\ \overline{P}_{RU_j}\end{bmatrix} = \begin{bmatrix}\alpha_i(\mathbb{P}^i(\boldsymbol{X},\boldsymbol{T})) & \alpha_i(\mathbb{P}^j(\boldsymbol{X},\boldsymbol{T}))\\ \alpha_j(\mathbb{P}^i(\boldsymbol{X},\boldsymbol{T})) & \alpha_j(\mathbb{P}^j(\boldsymbol{X},\boldsymbol{T}))\end{bmatrix} \cdot \begin{bmatrix}\overline{R}^i(\mathbb{P}^i(\boldsymbol{X},\boldsymbol{T}))\\ \overline{R}^j(\mathbb{P}^j(\boldsymbol{X},\boldsymbol{T}))\end{bmatrix} + \begin{bmatrix}\overline{P}^i_{stat}(\mathbb{P}^i(\boldsymbol{X},\boldsymbol{T}))\\ \overline{P}^j_{stat}(\mathbb{P}^j(\boldsymbol{X},\boldsymbol{T}))\end{bmatrix}, \quad (7)$$

with $(i,j) \in \mathcal{I}$, where $\alpha_i(\mathbb{P}^j(\boldsymbol{X},\boldsymbol{T}))$ representing the model slope of RU_i, using the CQI distribution of RU_j, and $\overline{P}^i_{stat}$

being the static consumed power of RU_i. This model thus corresponds to 2 planes, whose intersection provides the decision threshold. An example is shown with pair $(3, 6)$ in Figure 5.

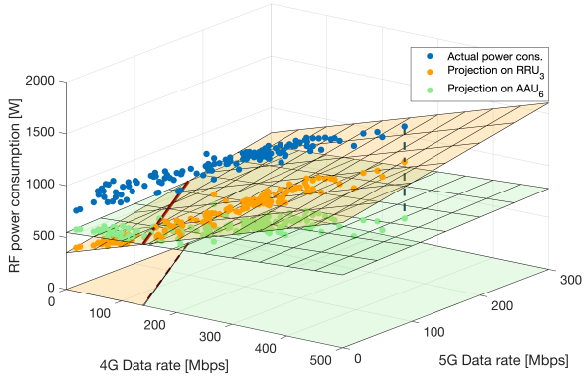

Figure 5. Power projection on models for RU pair (RRU_3, AAU_6).

The blue scatter points represent the actual total consumed power where both RUs are active, while orange and green points represent its projection on the 4G and 5G RU models, respectively. The intersection between these two model planes is given by the red straight line. It corresponds to the decision threshold locus, which separates the domain in two regions. The green (respectively orange) region indicates where is favorable from an energy point of view to reroute users to a 5G-AAU (respectively a 4G-RRU).

B. Results

The results of this implemented mechanism on the pair $(3, 6)$ are illustrated on Figure 6 and Figure 7. Figure 6 corresponds to the current situation, where both RUs are active, showing the average data traffic on each RU and its contribution to the total power consumption over the week on an hourly basis. Figure 7 is obtained applying the mechanism described above. Colors indicate the equipment to which the total data traffic is redirected. It shows a significantly reduced power consumption with the 5G-RU active most of the time. The only time the 5G-RU is switched-off in favor of the 4G-RU is at night, i.e., between 3 and 8 a.m. Note that the AFE's contribution is also visible for both technologies, and that the power consumption remains relatively flat, thanks to the higher energy-efficiency of the 5G-AAU.

The last step aims to benchmark the above power saving mechanism with other deactivation scenarios by quantifying the energy savings for the entire BS over one week. For that purpose, several scenarios are considered in the benchmark:

- Actual: current situation where all RUs are running,
- Scenario 1: hourly deactivation of 4G-LTE RUs and redirecting data traffic on 5G-NR RUs, if total data traffic reaches 80% of the maximum capacity of the 4G-LTE RU. Maximum capacity can be computed using (5) and considering N_{PRB}^{max},
- Scenario 2: hourly deactivation of 4G-LTE RUs and redirecting all data traffic on 5G-NR RUs,

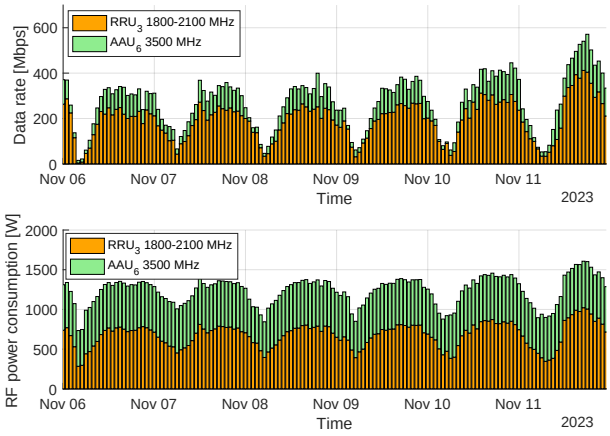

Figure 6. Power and total average data rate vs. time over a week in current situation.

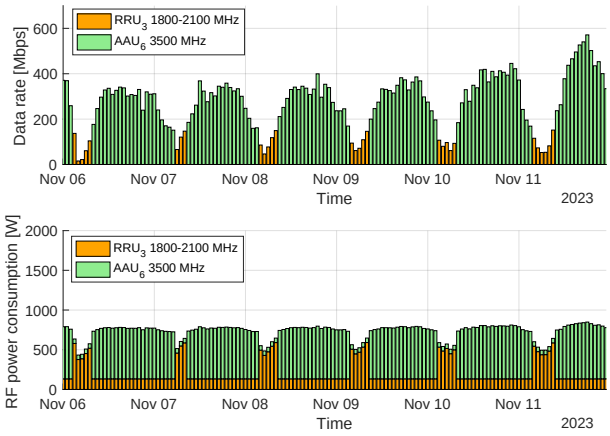

Figure 7. Power and total average data rate vs. time over a week when applying Scenario 3.

- Scenario 3: hourly deactivation of 4G-LTE RUs or 5G-NR RUs based on minimum power threshold criterion, as described previously.

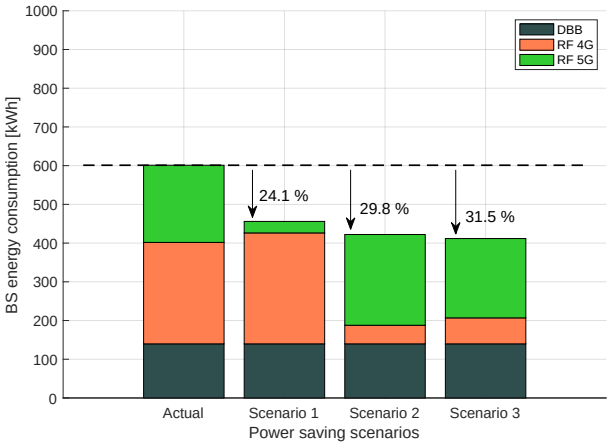

Figure 8. Benchmark of the energy savings for the entire BS over a week for different scenarios.

The results are given in Figure 8. Scenario 1 already shows a

power reduction of 24.1% with 4G-LTE almost always active, meaning that it rarely reaches 80% of its maximum capacity. Scenario 2 provides an even larger power reduction with 29.8% using 5G-NR only. Finally, Scenario 3 shows a slightly lower power consumption of 31.5% by switching between 4G-LTE and 5G-NR, following the methodology described above.

VI. Conclusion and Future Work

This paper presents an on-site analysis of power consumption of radio equipment, providing a methodology for reducing the power usage by dynamically deactivating RUs based on the average data traffic on an hourly basis. Our findings reveal that 5G-NR RUs demonstrate 3 to 9 times higher dynamic energy efficiency compared to its 4G counterparts.

The core contributions of this study include the extension of existing power consumption models as a function of the downlink data rate. Those models are used to provide a time-scheduling for equipment use and shows that 5G-NR equipment should be privileged at higher traffic levels, i.e., $\geq$ 150 Mbps, while 4G-LTE equipment is preferable at lower levels, due to their smaller static power consumption. This work also proposes three scenarios for RU deactivation: 1) using 4G-LTE RUs only and redirecting the data rate on 5G-NR RUs when reaching 80% of maximum 4G-LTE capacity, 2) redirecting to 5G-NR RUs only, and 3) selecting between 4G-LTE and 5G-NR RUs based on the derived power consumption vs. data rate model. Our results indicate that Scenario 3 offers the largest power savings, achieving a 31.5% reduction in power consumption for the entire base station over a week. In contrast, the current situation is the most energy-consuming, when both technologies are used simultaneously.

These findings attest the potential for considerable energy savings using deployed radio equipment, while maintaining a comparable channel quality. The implications of our work are significant for current radio access network systems, where implementing intelligent RU deactivation may help mobile operators reducing energy and carbon footprints of their RAN, as well as their operational costs, without compromising user's QoS. Measurements and discussions with mobile operators reveal that some pieces of equipment are already partially deactivated during nighttime. All these results however assume that all UEs are both 4G-LTE and 5G-NR compatible, which is not yet the case in Belgium.

Future work should also consider other QoS metrics such as latency, as well as quantify the gain margin from integration of additional power-saving features, e.g., with lower time granularity. Finally, it is crucial to validate the assumption regarding the conservation of the channel quality for users under different deactivation scenarios.

Acknowledgment

The authors would like to thank MWingz for their support.

References

[1] D. Lundén, J. Malmodin, P. Bergmark, and N. Lövehagen, "Electricity Consumption and Operational Carbon Emissions of European Telecom Network Operators," *Sustainability*, vol. 14, no. 5, p. 2637, 2022.

[2] L. Stobbe *et al.*, *Umweltbezogene Technikfolgenabschätzung Mobilfunk in Deutschland (Environmental Technology Assessment of Mobile Communications in Germany)*, Umweltbundesamt, 2023.

[3] L. Golard, J. Louveaux, and D. Bol, "Evaluation and projection of 4G and 5G RAN energy footprints: The case of Belgium for 2020–2025," *Ann. Telecommun.*, vol. 78, pp. 313–327, 2023.

[4] "Green 5G: Building a sustainable world," Analysys Mason, Tech. Rep., Aug. 2020.

[5] C. Freitag *et al.*, "The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, 2021.

[6] T. Islam, D. Lee, and S. S. Lim, "Enabling Network Power Savings in 5G-Advanced and Beyond," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1888–1899, 2023.

[7] D. López-Pérez *et al.*, "A survey on 5G radio access network energy efficiency: Massive MIMO, lean carrier design, sleep modes, and machine learning," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 653–697, 2022.

[8] L. Golard *et al.*, "A Parametric Power Model of Multi-Band Sub-6 Ghz Cellular Base Stations Using On-Site Measurements," in *Proceedings of the 2024 IEEE 35th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, IEEE, 2024, pp. 1–7.

[9] F. Launay and A. Perez, *LTE Advanced Pro: Towards the 5G Mobile Network*. John Wiley & Sons, 2019.

[10] B. Debaillie, C. Desset, and F. Louagie, "A Flexible and Future-Proof Power Model for Cellular Base Stations," in *IEEE 81st Veh. Technol. Conf.*, 2015.

[11] C. Desset, P. Wambacq, Y. Zhang, M. Ingels, and A. Bourdoux, "A flexible power model for mm-wave and thz high-throughput communication systems," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, 2020, pp. 1–6.

[12] A. Capone, S. D'Elia, I. Filippini, A. E. Redondi, and M. Zangani, "Modeling energy consumption of mobile radio networks: An operator perspective," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 120–126, 2017.

[13] 3GPP Technical Specification TS 36.213 V18.1.0, *Radio access network; Evolved universal terrestrial radio access (E-UTRA); Physical layer procedures (release 18)*, 2023.

[14] 3GPP Technical Specification TS 38.306 V18.0.0, *Radio access network; NR; User Equipment radio access capabilities (release 18)*, 2023.

[15] 3GPP Technical Specification TS 38.214 V18.1.0, *Radio access network; NR; Physical layer procedures for data (release 18)*, 2023.

[16] F. J. Martín-Vega, J. C. Ruiz-Sicilia, M. C. Aguayo, and G. Gómez, "Emerging tools for link adaptation on 5G NR and beyond: Challenges and opportunities," *IEEE Access*, vol. 9, pp. 126 976–126 987, 2021.

[17] E. Peralta, G. Pocovi, L. Kuru, K. Jayasinghe, and M. Valkama, "Outer loop link adaptation enhancements for ultra reliable low latency communications in 5G," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*, IEEE, 2022, pp. 1–7.

[18] F. Rezaei, M. Hempel, and H. Sharif, "LTE PHY performance analysis under 3GPP standards parameters," in *2011 IEEE 16th international workshop on computer aided modeling and design of communication links and networks (CAMAD)*, IEEE, 2011, pp. 102–106.

[19] Y. Wang, W. Liu, and L. Fang, "Adaptive modulation and coding technology in 5g system," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2020, pp. 159–164.