Flexcomm Simulator: Exploring Energy Flexibility in Software Defined Networks with ns-3

Rui Pedro C. Monteiro
HASLab, INESC TEC
University of Minho
Braga, Portugal
rui.p.monteiro@inesctec.pt

João Marco C. Silva
HASLab, INESC TEC
University of Minho
Braga, Portugal
joaomarco@di.uminho.pt

ABSTRACT

The digitalization of energy generation and distribution systems opens new opportunities for devising network operation and traffic engineering strategies capable of adapting to the energy availability and sources. Despite the potential, developing and testing new approaches are challenging in production environments. Furthermore, no simulators support such integration between the communication infrastructure and the power grid. Thus, this paper introduces *Flexcomm Simulator*, a tool based on ns-3 that supports developing and assessing multiple strategies toward green networking and communications driven by real-time information from the power grid (*i.e.*, *Energy Flexibility*). The proof-of-concept results demonstrate this contribution's potential by implementing an energy-aware routing algorithm that adapts to real-world *Energy Flexibility* data in a Metropolitan Area Network (MAN). Also, it showcases the simulator's capacity to deal with large-scale simulations through MPI-based distributed environments.

CCS CONCEPTS

• **Networks** → **Network simulations**; **Routing protocols**; **Traffic engineering algorithms**; • **Hardware** → **Renewable energy**.

KEYWORDS

Energy Flexibility, ns-3, Simulation, Software Defined Networking

ACM Reference Format:
Rui Pedro C. Monteiro and João Marco C. Silva. 2023. Flexcomm Simulator: Exploring Energy Flexibility in Software Defined Networks with ns-3. In *2023 Workshop on ns-3 (WNS3 2023), June 28–29, 2023, Arlington, VA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3592149.3592160

1 INTRODUCTION

The continuous growth of the Information and Communication Technologies (ICT) sector has led it to account for 5% to 9% of the global electricity consumption and more than 2% of greenhouse gas emissions [13, 22]. Communication networks alone are responsible for half of these amounts [16]. This scenario will be further stressed when considering the worldwide adoption of 5G technology and IoT-based systems and the upcoming 6th generation of mobile networks [11]. Therefore, the way massive amounts of data are transmitted is crucial to reducing operational costs and pursuing environmentally sustainable communication infrastructures [3, 14].

Within this context, network infrastructure optimisation has become a central research topic, with recent advances fostered by the increasing deployment of Software Defined Networking (SDN). This technology introduces a new paradigm of network management based on dynamic and programmatic network configuration in real-time. Most SDN-based research on energy efficiency focuses on new techniques toward efficient traffic engineering [25] combined with selectively turning off devices during off-peak periods [7]. However, the latter approaches are usually only aimed at reducing operating costs while maintaining acceptable levels of Quality of Service (QoS), with no focus on environmental impact.

The energy market is also undergoing a comprehensive transformation driven by the digitalization of infrastructures and processes. These modern energy grids (*i.e.*, Smart Grids) aim primarily for intelligent and autonomous operations, but they can also provide real-time information about their state to external systems. This information, called *Energy Flexibility* [19], includes energy availability in specific regions, target consumption, and energy sources, which SDN *controllers* can exploit to select efficient infrastructure operations or traffic engineering strategies dynamically.

Since modern Distributed Energy Resources (DERs) are affected by the intermittent nature of renewable sources, innovative and sustainable strategies based on *Energy Flexibility* will target mainly large-scale distributed networks. Despite being a promising research line, the underlying developments and evaluation in production environments face significant challenges as operators must ensure high performance to comply with QoS requirements.

To cope with these challenges, this paper presents the Flexcomm Simulator[1]. This ns3-based simulation environment introduces a comprehensive platform to support developing and assessing energy-aware strategies driven by the integration of SDN and *Energy Flexibility* data. By providing a user-friendly tool to define network topologies, workload scenarios, and energy constraints, allied with relevant statistical indicators, the Flexcomm Simulator can foster research and innovation on sustainable networks.

The paper is structured as follows: Section 2 introduces SDN and *Energy Flexibility* related concepts and some related work. Section 3 presents the Flexcomm Simulator and its implementation details. Section 4 discusses promising early work in new energy flexibility-aware routing algorithms being developed with the simulation tool.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WNS3 2023, June 28–29, 2023, Arlington, VA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0747-6/23/06…$15.00
https://doi.org/10.1145/3592149.3592160

[1] Available at https://github.com/RuiCunhaM/Flexcomm-Simulator

Finally, Section 5 concludes on the final results and future work perspectives.

2 BACKGROUND AND RELATED WORK

Although research on *green networking and communications* has intensified over the last decade, most developments were focused on wireless networks or data centers. In geographically distributed wired networks, the most common strategy consists of turning off part of the devices during off-peak periods without considering sustainability-related aspects [7]. However, the recent growth of SDN infrastructures and the ongoing digitization of power system infrastructures and processes have opened up the opportunity to develop more complex strategies, for example, prioritizing network regions powered by renewable sources or with higher availability intervals. Such strategies are now possible by integrating the SDN control plane with *Energy Flexibility* Application Programming Interfaces (APIs).

Software Defined Networking is a dynamic and manageable approach to network management. This architecture addresses the static nature of conventional networks mainly by decoupling the control and data planes [20]. To achieve this, it centralizes the control plane while maintaining an open interface between the devices of both planes. As shown in Figure 1, this new paradigm differs from traditional networks by possessing one or more *Controllers* with a global network view. These entities can, through a *Southbound API*, update the forwarding rules of network devices, dynamically changing the network behavior. The controllers can also receive instructions from applications along a *Northbound API*, providing programmable user access to networks. Furthermore, classical routers are replaced with similar devices striped from their control plane, only implementing forwarding functions. These devices are usually designated as programmable switches or SDN switches, but for simplification purposes, they are just addressed as switches for the remainder of this paper.

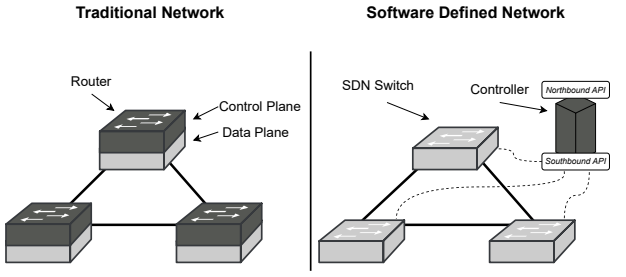

Figure 1: Traditional Network *vs* Software Defined Network

Although new technologies have emerged in recent years (*e.g.*, Programming Protocol-independent Packet Processors (P4) [4]), OpenFlow [17] is still regarded as a reference within the SDN context, defining a communication protocol between controllers and switches. Using this standard channel, the controller can monitor the device and its port status, but most importantly, it can configure switches' flow table entries with fine-grained control.

Previous works have proposed new traffic engineering strategies in SDN to help reduce energy consumption [25]. However, these techniques mainly focus on turning off devices. Although selectively shutting down equipment has proven to be an effective method to reduce the total energy consumed by networks [7–9], these strategies are only suitable during low traffic demand periods. Therefore, new traffic engineering techniques are still required to address broader scenarios and foster enhancements in large-scale networks' overall energy consumption.

Energy Flexibility is a fairly new concept which has gained significant interest following the evolution of power grids in recent years. The fundamental idea behind modern Smart Grids' flexibility is the ability to adjust energy supply and demand to maintain balance dynamically [19]. This is important because in DERs, energy supply may vary independently across multiple grid points. At the same time, energy consumption can also fluctuate following weather conditions and the different demands along the day or week [10].

Based on the fact that some producers and demanding consumers can actively change generation and demand trends (*e.g.*, flexible loads, controllable local generation), *Energy Flexibility* can potentially drive the equilibrium of power grids. Moreover, by following recommended consumption and production values, some consumers and producers might temporarily compensate others incapable of doing the same. In return, by sticking to this mechanism, consumers can benefit from incentives and reduced prices.

For distributed and high-demanding networks (*e.g.*, Internet Service Provider - ISP), *Energy Flexibility* represents economic and sustainability opportunities. By changing routing policies, networks can shift their computational load across the infrastructure, targeting geographical areas with higher flexibility, thus, benefiting from reduced prices. From a sustainability point of view, more flexibility can also reflect energy produced by renewable sources. Therefore, prioritizing these areas also contributes to reducing the environmental impact.

Aiming to foster these integrated systems, Universal Smart Energy Framework (USEF) introduced the Flexibility Value Chain (FVC), which defines the role of each participant in the power grid and their interaction protocols [24]. The main intervening in this system is the *Aggregator*. As a central entity, the *Aggregator* is responsible for coordinating the flexibility of all energy generators. Based on predictable energy consumption and generation models, at this point, high-demanding consumers get flexibility-related data for multiple geographical areas on a day-ahead scheme. Such information is provided through an open standard API.

Despite the significant potential, no previous work has focused on these strategies due to the challenges of testing them in production environments and the lacking of simulation tools capable of providing the components required to develop and assess new approaches.

3 FLEXCOMM SIMULATOR

The main goal of Flexcomm Simulator is to create a simulation environment to support developing and evaluating energy-efficient strategies that combine SDN and *Energy Flexibility* to reduce the environmental impact of network infrastructures. The platform must be able to realistically recreate network infrastructures and their component characteristics (*e.g.*, devices' energy consumption).

Moreover, network traffic must also be simulated accurately, ensuring that changes in QoS can be assessed appropriately when developing new routing algorithms. It is also essential to guarantee a user-friendly interface through which operators and researchers can promptly develop new traffic engineering techniques. Lastly, the tool should accommodate large-scale network simulations.

Based on these design goals, the primary contribution of Flexcomm Simulator is providing scalable ways to accurately compare the impact of different engineering strategies on energy consumption and QoS in multiple topologies and traffic conditions. This is achieved through a *pipeline workflow* illustrated in Figure 2. Each simulation scenario is defined as a set of configuration files later parsed by the tool and converted in the respective ns-3's simulation. During runtime, multiple logging files containing details on the network's status are generated to be further processed to compile the statistics revealing how the network performed. This simple approach represents a flexible method to iterate over different routing strategies and distinct test scenarios, allowing to put new and existent algorithms to test across diverse simulated conditions. Furthermore, this architecture reduces the amount of programming required since only the routing logic needs to be implemented in SDN controllers.

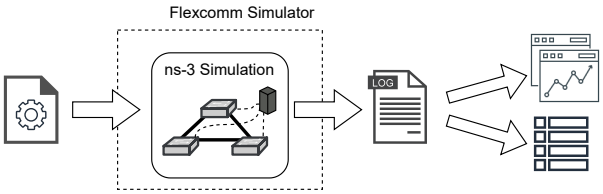

Figure 2: Flexcomm Simulator Workflow

3.1 Architecture Overview

The Flexcomm Simulator is built with ns-3.35 at its core, combining some modified third-party libraries with newly proposed components as depicted in Figure 3. The main simulation logic behind SDN is provided by the OFSwitch13 module [6], which enhances ns-3 with OpenFlow 1.3 support. The implementation includes a detailed software switch, followed by an extendable controller interface that can be used to implement any routing policy. Additionally, the *EnergyAPI* component, proposed by this work, provides *Energy Flexibility* data to the simulation, allowing SDN controllers to access information about the energy available for each switch.

To generate simulation statistics, four distinct components calculate and report information about the network elements. Switches' energy consumption is estimated by ECOFEN, an energy framework designed to evaluate power consumption in large-scale wired networks [21]. This framework's energy models allow the simulation of several real energy consumption profiles. The *SwitchStats* and *LinksStats* components, also proposed by this work, act as probes measuring the workload of each switch and link, respectively. Lastly, the *FlowMonitor* [5] also works as a set of probes classifying each simulated network flow and reporting different performance metrics.

This work also introduces a *Parser* component, designed to process the configuration files describing the simulation scenarios and to create the correspondent ns-3's simulation without requiring any C++ programming.

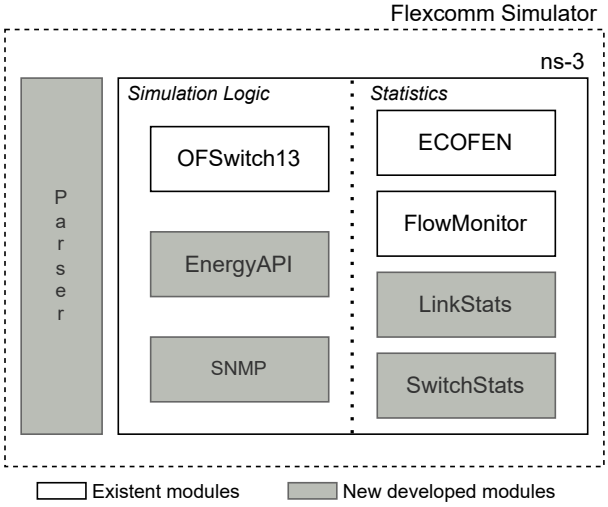

Figure 3: Flexcomm Simulator Components

Each simulation is composed of five main elements, namely, *hosts*, *switches*, *links*, *applications*, and *controllers*. *Switches*, *controllers* and *links* map directly to their real-life representation. Meaning that these are the network devices that together constitute a network and its administrative domain. The *hosts*, on the other hand, are abstractions required by ns-3. They do not necessarily represent an end device in the simulation, as the name would indicate. In some simulation scenarios, even if just the core of a network is being replicated, a host is still required on each network end from where traffic can be addressed or originate. This happens because *applications* that are used to generate and receive simulated network traffic can only be part of a *host*.

3.2 Parser

A new configuration format is included alongside the Flexcomm Simulator to provide a simple syntax to define simulations. The new format not only allows describing the network topology but also generic simulation configurations in a non-verbose syntax and independently on knowledge of ns-3's internal components. This ensures better compatibility since configuration files do not become unusable with internal APIs changes. Moreover, an agnostic configuration format can be ported to other tools to replicate similar experiments.

Each simulation session requires four distinct Tom's Obvious Minimal Language (TOML) configuration files. The *nodes.toml* and *links.toml* describe the network topology by specifying its nodes (*e.g.*, switches, hosts) and links, respectively. The *applications.toml* holds the different network traffic generators and underlying distribution. At last, *configurations.toml* contains generic simulation configurations, for example, the simulation duration and the different types of data that should be logged. More comprehensive details

on how each simulation is configured are available at Flexcomm Simulator's Documentation[2].

To handle the new configuration format, the *Parser* component is responsible for reading the input files and deploying the correspondent ns-3's simulation. As the main program entry point, this component orchestrates the coordination between the different simulation elements, translating all the configurations in the TOML files to ns-3 specific logic. Additionally, it deals with all the necessary integrations and connections between different modules required in a ns-3's simulation.

3.3 OFSwitch13

OFSwitch13 is a module developed to enhance ns-3 with OpenFlow version 1.3 and an extensible controller interface [6]. Both devices communicate using real OpenFlow messages sent through the ns-3 protocol stack. Additionally, two types of SDN controllers can interact with the simulation. Besides internal controllers that can be developed directly into ns-3 and resorting to C++, the module also includes experimental support for real SDN controllers (*e.g.*, ONOS, Ryu), external to the ns-3 simulation. This functionality is achieved by running simulations in real-time and connecting ns-3 to a non-simulated external SDN controller using Linux Tap Bridges. Such implementation has been validated within the Flexcomm Simulator, which also allows the use of external SDN controllers, making it possible to test and validate routing strategies developed with real application stacks.

Contrary to internal controllers, external SDN controllers cannot access the simulated device status using ns-3 C++ methods. Thus, based on the extensibility support of OpenFlow, this paper also proposes new messages and methods to fetch such information from the base simulator. The extension, included within the Flexcomm Simulator, allows external controllers to obtain Central Process Unit (CPU) usage and energy consumption values from simulated programmable switches through two new message types.

3.4 ECOFEN

ECOFEN framework adds different energy models to ns-3 [21] aiming to support assessing energy consumption in multiple network devices. These models can be configured with any real energy consumption profile. The ones already included for the device's chassis and network interfaces range from a simpler behavior, where consumption is only calculated considering the time a device spends turned ON or turned OFF, to more complex ones, capable of taking into account the energy spent to process each byte.

Although accurate, the complex models require detailed measurements not typically available or disclosed by device manufacturers. Thus, in addition to supporting the included energy models, the Flexcomm Simulator also introduces a new and more generic model based on the device's computational load. With this implementation, the energy consumption is estimated as a function of the switch's CPU usage reported by the OFSwitch13 module. This allows configuring discrete energy consumption values for different workloads, a standard method to measure devices' energy consumption.

Moreover, further changes are proposed to ECOFEN addressing its energy reports limitations. By default, energy estimations are written to the *stderr* during a simulation. Besides being inefficient, this mechanism is also unsuitable for the Flexcomm Simulator approach of generating logging files for later analysis. Thus, changes are applied to the framework, allowing energy consumption to be logged into output files in a more compact format and at configurable time intervals.

[2]Available at https://github.com/RuiCunhaM/Flexcomm-Simulator-User-Manual

3.5 Energy API

Bringing the previously mentioned *Aggregator's* API into the simulation environment requires a new component to recreate such a system. The aggregator defines a target consumption based on previous records and the energy availability for a time window. It then exposes the predicted energy consumption and the relative flexibility to consumers through a Representational State Transfer (REST) API. The information is made available as two distinct fixed-size vectors, where each position corresponds to an equal time window for the next 24 hours.

To expose this behavior in simulations, the *EnergyAPI* module reads two additional JavaScript Object Notation (JSON) configuration files, each containing energy consumption estimation and *Energy Flexibility*, respectively. As shown in Figure 4, depending on the type of controller being used in the simulation, *EnergyAPI's* data can be accessed differently. Internal controllers can obtain *Energy Flexibility* information using method invocations and providing the location/device identifier. External controllers can access an identical API, available through a Hypertext Transfer Protocol (HTTP) server, more realistically resembling the real one. This server reads the same JSON input files as the simulator and makes the endpoints available through *localhost*. When an external controller is connected to the simulator, the external HTTP server is automatically started as a *child* process of the main simulator.

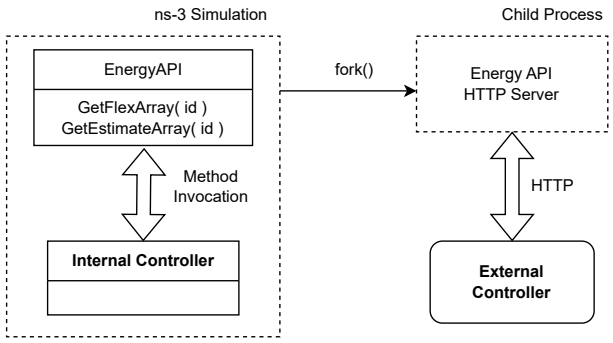

Figure 4: Energy API Architecture

3.6 SNMP

Although the proposed OpenFlow extension provides external controllers richer information about switch operation (see Section 3.3), Simple Network Management Protocol (SNMP) [12] is still a more common and established method of fetching device-related data. It uses unique identification (*i.e.*, Object IDentifiers (OIDs)) to manageable objects that are grouped in Management Information Bases (MIBs). This identification allows remote entities to get device-related information and change their configurations. Besides some

established and standardized MIBs, typically, each manufacturer creates its own, leading to interoperability challenges.

To accommodate real SDN controllers that resort to this protocol, the Flexcomm Simulator introduces a prototype SNMP implementation for ns-3. This support is achieved with an auxiliary simulator (*i.e.*, *snmpsim*[3]) that acts as multiple devices and responds to SNMP requests sent by controllers. Similarly to the *EnergyAPI*, the auxiliary simulator is started as a child process of the main simulator when an external controller is connected to the simulation.

3.7 Traffic Generators

An essential part of achieving accurate network simulations is the ability to simulate realistic traffic profiles. To generate simulated network traffic, ns-3 provides internal applications capable of modeling simple behavior patterns. In addition to these applications, the Flexcomm Simulator supports extra traffic generators better suited for large distributed networks (*e.g.*, Metro, Core), namely *PPBP* [1] and *SINGen*. Ammar et al. introduced the *PPBP* traffic generator based on the Poisson Pareto Burst Process (PPBP) [26], a model developed as a simple representation of aggregated Internet traffic. It replicates multiple overlapping constant bursts that follow a heavy-tailed distribution.

The *SINGen* generator is proposed by this paper and is also included with the simulation tool to model traffic patterns that closely match sinusoidal functions [7, 15, 18, 23]. These patterns are commonly observed as a result of citizens' daily routine that consequently creates peak and off-peak periods nominated Tidal Effects [23]. Additionally, all generators can be configured with different options to fit the intended traffic profiles. The respective required sinkers that act as the receiving end for the generated traffic are automatically handled by the *Parser* component, with no associated configurations required.

3.8 Statistics

In addition to energy consumption estimations, the Flexcomm Simulator also generates statistics to help analyze how different routing strategies affect QoS. This data focus on providing insights into the status of the network's elements and the performance impact in network communications. Metrics conveying the status of devices allow assessing how the overall computational load is distributed across the network, while changes in network flows (*e.g.*, delay, packet loss) translate the changes in QoS experienced by communications.

A new module called *SwitchStats* is in charge of calculating the metrics associated with each switch's utilization. For each switch, the module reports in regular time intervals its CPU usage, the number of packets processed and the number of bytes processed on those packets. The CPU utilization is obtained directly from the OFSwitch13 module, that based on a configurable switch capacity can estimate its workload depending on the data rate traversing that switch at any given moment. As for the number of packets and bytes, the *SwitchStats* module resorts to ns-3's tracing system, binding callbacks that listen and keep track of incoming packets.

Similarly, the *LinkStats* module also resorts to ns-3's tracing system to monitor, for each link, the utilization, number of packets and number of bytes processed. Through the configuration files, both modules can independently be activated or deactivated depending on the intended data to be collected. Moreover, the frequency used to collect statistics can also be changed, allowing for customizable logging intervals.

To inspect how network flows are affected by changes in routing policies, the Flexcomm Simulator utilizes the FlowMonitor module [5]. This component also leverages ns-3's tracing system to install probes into simulated nodes, listening for exchanged packets between them. During the runtime, the FlowMonitor module classifies the network flows and calculates various statistics on each one. Processing an Extensible Markup Language (XML) file generated at the end of a simulation yields multiple IP Performance Measurement (IPPM) metrics about each individual network flow simulated. These metrics allow diagnosing changes in communication's delay and consequent *jitter*, as well as packet losses that can be induced by changes in routing logic. Like other data sources, the FlowMonitor data can be disabled from the configuration files. Furthermore, to help inspect changes in network flows, the Flexcomm Simulator allows to individually enable Packet Captures (PCAPs) for each link, providing an additional source of data.

3.9 Distributed Simulations

In order to improve the time it takes to run simulations, ns-3 includes support for Distributed Simulations. This allows it to take advantage of multicore processors or High Performance Computing (HPC) infrastructures, a valuable feature to help the simulation tool dealing with large simulation scenarios. The base functionality is guaranteed by the Message Passing Interface (MPI) implementation, where topologies can be partitioned across distinct Logical Processes (LPs), splitting the computational load across them. In this implementation, each LP/MPI rank is in charge of normally simulating its assigned portion of the network. When two nodes that belong to different ranks are required to communicate, the messages are exchanged using MPI. To prevent desynchronization across LPs, a different internal simulator implementation is used resorting to lookahead algorithms.

At the moment, ns-3's MPI implementation only works with Point-to-Point Protocol (PPP) links. This means that nodes belonging to different LPs can only be connected through this type of link. This limitation spreads to the OFSwitch13 module, preventing its compatibility with distributed simulations.

Currently, OFSwitch13's switches only expect connections using the Carrier-Sense Multiple Access (CSMA) link model since it is the only one that encapsulates network packets into Ethernet frames. Addressing such limitation, this paper proposes a new link model based on the existent PPP implementation, that instead utilizes Ethernet frames. The new model enhances ns-3 with a modern full-duplex Ethernet link abstraction that is also compatible with MPI. Furthermore, the integration of this link model into OFSwitch13 allows large-scale SDN to be simulated in distributed environments.

These changes have been validated within the Flexcomm Simulator, which fully supports distributed simulations. Through the configuration files, any simulated network topology can be partitioned across the desirable number of MPI ranks. This functionality is however limited to internal OFSwitch13 controllers, given that

[3]Available at https://github.com/inexio/snmpsim

real-time simulation required by external controllers utilize a simulator implementation that differs from the distributed one.

4 PROOF OF CONCEPT

Two proof of concepts are presented to demonstrate Flexcomm Simulator's potential to foster new energy-aware routing algorithms for SDN. The first showcases promising early work in energy flexibility-aware routing strategies developed with the proposed tool. These strategies demonstrate the ability of prioritizing energy availability while preserving networks' QoS. The second one focuses on demonstrating the capacity to deal with large-scale simulated wired networks by leveraging distributed simulations, thus, improving the time required to run each simulation.

4.1 Energy Flexibility-Aware Routing

As previously mentioned, *Energy Flexibility* can change over time depending on multiple factors, *e.g.*, fluctuating demands or variable productions from DERs. Independently of the cause, these values behave differently across geographical areas and consequently across the electrical grid. In network topologies that expand over large areas, energy can be more abundant in certain zones than in others. Good examples are MANs.

To compare how purposely designed energy flexibility-aware routing algorithms behave against more classic approaches, a simulation scenario based on Atlanta's MAN topology [2], composed by 15 switches and shown in Figure 5, is devised. In this simulation scenario, the network faces a Tidal Effect where the traffic demand increases gradually over time across the infrastructure.

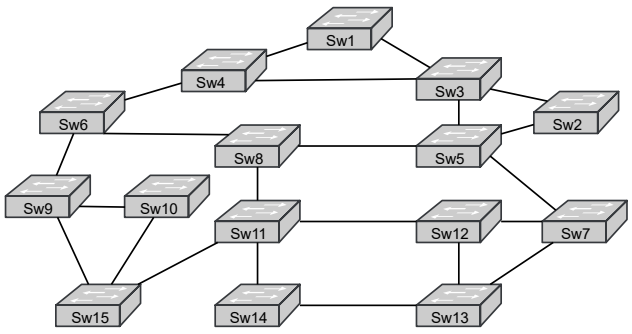

Figure 5: Atlanta's MAN Topology

At the same time, *Energy Flexibility* changes independently for different devices. Energy availability profiles for this particular simulation are generated as a combination of solar and wind energy production patterns publicly available at *Redes Energéticas Nacionais* (REN)' Data Hub [4]. However, other similar available datasets can be utilized to derive distinct flexibility profiles considering different energy and data sources. The same simulation scenario is evaluated using Open Shortest Path First (OSPF) as a traditional routing algorithm and an experimental energy flexibility-aware approach.

The results presented in Figures 6 and 7 show, for switches *Sw14* and *Sw5* respectively, the average energy consumption in 15 minutes time slots across a simulated 12 hours period. Each time slot matches the same time window used by the *EnergyAPI* to disclose *Energy Flexibility*. The plotted recommended energy consumption corresponds to the energy consumption estimate, plus the *Energy Flexibility*, setting a suggested maximum value target for what a given switch should consume.

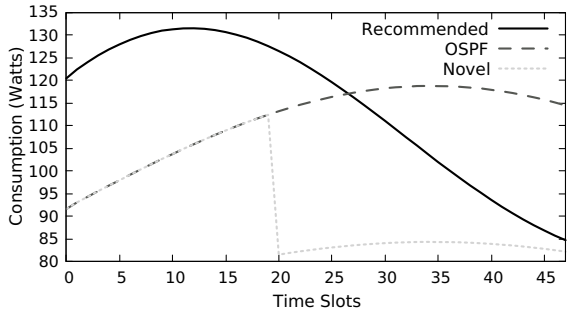

Figure 6: Switch Sw14 Energy Consumption

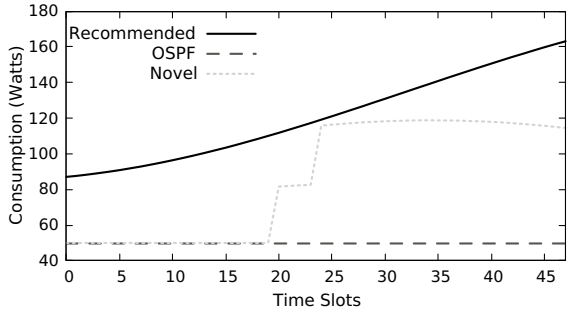

Figure 7: Switch Sw5 Energy Consumption

As Figure 6 shows, during the simulated period, the recommended energy available to switch *Sw14* decreases over time. When using OSPF to apply the routing rules, the algorithm is not capable of reacting to such an event, and eventually, the switch ends up operating above the suggested energy consumption limit. Meanwhile, the novel algorithm being developed minimizes this effect by anticipating its occurrence. Considering the *Energy Flexibility* and by extrapolating the current traffic demand, the algorithm predicts the eventual crossover point and takes action to prevent it. Around time slot number 20, as also seen in Figure 6, switch *Sw14*'s energy consumption drops to an inferior value, reflecting the workload decrease. Such happens because the controller reacts to the decrease in availability and preemptively shifts some workload away from the device, relocating network flows to alternative paths with more energy at their disposal.

In contrast, Figure 7 illustrates an increase in *Energy Flexibility* for switch *Sw5*. Again, OSPF cannot adapt to such changes, and the switch remains idle during the entire period. With the experimental routing logic, however, this increase in available energy is leveraged, and flows are gradually moved towards this device

[4] https://datahub.ren.pt/

when the algorithm detects that such action is possible while simultaneously relieving other switches.

Table 1 contains the averaged IPPM metrics from the simulated network flows, computed by the *FlowMonitor* module, and recorded when utilizing OSPF and the experimental approach as routing algorithms. The results show that when applying the energy flexibility-aware routing algorithm, there is an increase in the mean and maximum delay experienced by the flows. Furthermore, a small increase in the maximum *jitter* can also be observed. These changes, however, are expected and can be explained due to relocated network flows. When assigning a new network path to a flow prioritizing *Energy Flexibility*, it is expected that a *longer* path is selected when compared to OSPF, thus increasing delay. When the change takes place, it also causes a *jitter* spike since the delay increases slightly. Nevertheless, no packet losses occur during the transition. For most communications, these small changes in QoS are acceptable, not representing a significant impact.

Table 1: IPPM Metrics Between OSPF and Novel Algorithm

Metric	OSPF	Experimental
Maximum Delay (ms)	10.00	13.50
Minimum Delay (ms)	10.00	10.00
Mean Delay (ms)	10.55	12.68
Maximum Jitter (ms)	0.00	3.00
Minimum Jitter (ms)	0.00	0.00
Mean Jitter (ms)	0.00	0.00
Packet Loss (%)	0.00	0.00

Even from a proof of concept, these results demonstrate how *Energy Flexibility* can be exploited to optimize energy consumption in distributed networks by moving traffic across alternative paths. Moreover, it shows that this can be achieved with minimal QoS penalties. Additionally, these conclusions only reflect early work on developing new algorithms made possible by the Flexcomm Simulator. Advancements are expected in more complex algorithms capable of prioritizing communications that are more susceptible to changes in QoS (*e.g.*, Voice over IP - VoIP).

4.2 Scalability

Flexcomm Simulator's support for distributed simulations provides a method to enhance the time required to compute large simulations. To demonstrate the effectiveness of this mechanism, a network topology composed of 20 programmable switches and 20 hosts is devised. Two distinct simulation scenarios are established with 10 identical *SINGen* applications generating network traffic. In *Scenario 1*, applications are installed, generating traffic between hosts at a distance of 2 hops, while in *Scenario 2*, traffic is exchanged between hosts further away. Tests are conducted at Minho Advanced Computing Center (MACC)'s *BOB* supercomputer[5]. In both scenarios, the number of MPI ranks increases while the network topology is also partitioned following the same scheme. Figure 8 depicts the obtained speedups in *Scenario 1* and *Scenario 2*, averaged from 5 simulations, against the optimal speedup.

[5]https://macc.fccn.pt/

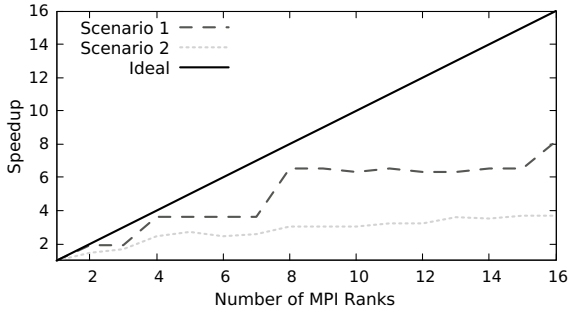

Figure 8: Simulation Speedup

Ideally, the speedup should increase proportionally to the number of MPI ranks sharing the total computational load of the simulation. In these tests, both scenarios demonstrate a reduction in simulation time. However, the real speedup is noticeably lower when compared with the optimal values. Furthermore, although the number of nodes and the total amount of traffic simulated is the same in both scenarios, *Scenario 1* clearly shows a better scaling when compared with *Scenario 2*. This behavior can be explained by ns-3's distributed simulations implementation and the need for an optimal topology partition that isolates communication clusters. In *Scenario 1*, given that hosts exchanging traffic are closer to each other, the number of LPs can further be increased while keeping communication clusters isolated in each network partition, thus, reducing MPI messages to a minimum. In contrast, *Scenario 2*'s traffic distribution prevents such isolation, requiring more intense usage of MPI communication to trade messages between LPs.

With this analysis, it is possible to conclude that the gains associated with distributed simulations heavily depend on the simulation scenario and how the network topology is partitioned. Nevertheless, this functionality is still a valuable resource in Flexcomm Simulator to improve the duration of large-scale network simulations and consequently fostering the development and evaluation of new routing strategies.

5 CONCLUSIONS AND FUTURE WORK

In this paper, the Flexcomm Simulator is proposed as a new simulation environment aiming to foster new energy-routing strategies for SDN focused on *Energy Flexibility*. The tool, built upon ns-3, puts together multiple features that enable new controller logic to be developed using the information provided by electrical grid operators. Moreover, the environment generates a comprehensive set of statistics that enable different strategies to be compared on their energy consumption and QoS impact. In addition to the simulator, the paper also showcases some early work in developing energy flexibility-aware routing algorithms capable of achieving a more balanced energy consumption. The experimental approach can adapt to changes in energy availability and relocate network flows across different regions of a network, respecting flexibility imposed by electrical grids while maintaining an acceptable QoS for communication networks.

For future work, besides the continuous updates of the Flexcomm Simulator with additional simulation features and to keep up with

changes in both SDN and *Energy Flexibility*-related technologies, the main focus is the development of new energy-aware routing strategies resorting to the simulation tool. With the simulation environment proposed by this paper, new routing algorithms leveraging *Energy Flexibility* information can be developed and properly evaluated more straightforwardly. Additionally, the future inclusion of the energy source into the information provided by the *Aggregator*, stating how the energy is produced, can also help to develop more complex techniques that target renewable sources, helping to reduce the environmental impact of network infrastructures.

ACKNOWLEDGMENTS

This work is financed by National Funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P. (Portuguese Foundation for Science and Technology) within the project FLEXCOMM, with reference EXPL/CCI-INF/1543/2021.

The authors acknowledge Minho Advanced Computing Center (MACC) for providing HPC resources that have contributed to the research results reported within this paper.

We also would like to thank João Pedro Meira for the help and insights in developing the OpenFlow extension.

REFERENCES

[1] Doreid Ammar, Thomas Begin, and Isabelle Guerin-Lassous. 2011. A New Tool for Generating Realistic Internet Traffic in ns-3. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques* (Barcelona, Spain). ACM. https://doi.org/10.4108/icst.simutools.2011.245548

[2] Mohamad Khattar Awad, Yousef Rafique, and Rym A. M'Hallah. 2017. Energy-aware Routing for Software-defined Networks with Discrete Link Rates: A Benders Decomposition-based Heuristic Approach. *Sustainable Computing: Informatics and Systems* 13 (2017), 31–41. https://doi.org/10.1016/j.suscom.2016.11.003

[3] Raffaele Bolla, Roberto Bruschi, Franco Davoli, and Flavio Cucchietti. 2011. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. 13, 2 (2011), 223–244. https://doi.org/10.1109/SURV.2011.071410.00073

[4] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014-07-28. P4: Programming Protocol-independent Packet Processors. 44, 3 (2014-07-28), 87–95. https://doi.org/10.1145/2656877.2656890

[5] Gustavo Carneiro, Pedro Fortuna, and Manuel Ricardo. 2009. FlowMonitor - A Network Monitoring Framework for the Network Simulator 3 (ns-3). In *Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools* (Pisa, Italy). ICST. https://doi.org/10.4108/ICST.VALUETOOLS2009.7493

[6] Luciano Jerez Chaves, Islene Calciolari Garcia, and Edmundo Roberto Mauro Madeira. 2016. OFSwitch13: Enhancing ns-3 with OpenFlow 1.3 Support. In *Proceedings of the Workshop on ns-3 - WNS3 '16* (Seattle, WA, USA). ACM Press, 33–40. https://doi.org/10.1145/2915371.2915381

[7] Luca Chiaraviglio, Marco Mellia, and Fabio Neri. 2009-06. Energy-aware Backbone Networks: A Case Study. In *2009 IEEE International Conference on Communications Workshops* (Dresden, Germany). IEEE, 1–5. https://doi.org/10.1109/ICCW.2009.5208038

[8] Luca Chiaraviglio, Marco Mellia, and Fabio Neri. 2009-06. Reducing Power Consumption in Backbone Networks. In *2009 IEEE International Conference on Communications* (Dresden, Germany). IEEE, 1–6. https://doi.org/10.1109/ICC.2009.5199404

[9] Luca Chiaraviglio, Marco Mellia, and Fabio Neri. 2012-04. Minimizing ISP Network Energy Cost: Formulation and Solutions. 20, 2 (2012-04), 463–476. https://doi.org/10.1109/TNET.2011.2161487

[10] European Commission, Directorate-General for Energy, Michelle Antretter, Marian Klobasa, Matthias Kühnbach, Mahendra Singh, Kaspar Knorr, Jonathan Schütt, Jordy de Boer, Diego Hernandez Diaz, Franziska Fitzschen, Andrés Garcerán, Ricardo Reina, Ole Rolser, Stefanie Stemmer, Jan Steinbach, and Eftim Popovski. 2022. *Digitalisation of Energy Flexibility.* Publications Office of the European Union. https://doi.org/doi/10.2833/113770

[11] Ericsson. 2022-02. 6G - Connecting a Cyber-physical World. (2022-02). https://www.ericsson.com/en/reports-and-papers/white-papers/a-research-outlook-towards-6g

[12] Mark Fedor, Martin Lee Schoffstall, James R. Davin, and Dr. Jeff D. Case. 1990. Simple Network Management Protocol (SNMP). RFC 1157. https://doi.org/10.17487/RFC1157

[13] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon S. Blair, and Adrian Friday. 2021. The Real Climate and Transformative Impact of ICT: A Critique of Estimates, Trends, and Regulations. *Patterns* 2, 9 (2021), 100340. https://doi.org/10.1016/j.patter.2021.100340

[14] Rahil Gandotra and Levi Perigo. 2021. A Comprehensive Survey of Energy-Efficiency Approaches in Wired Networks. 261–282. https://doi.org/10.5121/csit.2021.111822

[15] José Luis García-Dorado, Alessandro Finamore, Marco Mellia, Michela Meo, and Maurizio Matteo Munafò. 2012-06. Characterization of ISP Traffic: Trends, User Habits, and Access Technology Impact. 9, 2 (2012-06), 142–155. https://doi.org/10.1109/TNSM.2012.022412.110184

[16] IEA. 2021. Data Centres and Data Transmission Networks. https://www.iea.org/reports/data-centres-and-data-transmission-networks

[17] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008-03-31. OpenFlow: Enabling Innovation in Campus Networks. 38, 2 (2008-03-31), 69–74. https://doi.org/10.1145/1355734.1355746

[18] Janine Morley, Kelly Widdicks, and Mike Hazas. 2018-04. Digitalisation, Energy and Data Demand: The Impact of Internet Traffic on Overall and Peak Electricity Consumption. 38 (2018-04), 128–137. https://doi.org/10.1016/j.erss.2018.01.018

[19] National Grid ESO. 2020. Introduction to Energy System Flexibility. https://www.nationalgrideso.com/document/189851/download

[20] ONF. 2012. Software-Defined Networking: The New Norm for Networks. (2012). https://opennetworking.org/sdn-resources/whitepapers/software-defined-networking-the-new-norm-for-networks/

[21] Anne-Cecile Orgerie, Laurent Lefevre, Isabelle Guerin-Lassous, and Dino M. Lopez Pacheco. 2011-06. ECOFEN: An End-to-end Energy Cost Model and Simulator for Evaluating Power Consumption in Large-scale Networks. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks* (Lucca, Italy). IEEE, 1–6. https://doi.org/10.1109/WoWMoM.2011.5986203

[22] Aimee Ross and Lorna Christie. 2022. Energy Consumption of ICT. (2022). https://post.parliament.uk/research-briefings/post-pn-0677/

[23] Sebastian Troia, Gao Sheng, Rodolfo Alvizu, Guido Alberto Maier, and Achille Pattavina. 2017-03. Identification of Tidal-traffic Patterns in Metro-area Mobile Networks via Matrix Factorization Based Model. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (Kona, HI). IEEE, 297–301. https://doi.org/10.1109/PERCOMW.2017.7917576

[24] USEF. 2018. Flexibility Value Chain. https://www.usef.energy/app/uploads/2018/11/USEF-White-paper-Flexibility-Value-Chain-2018-version-1.0_Oct18.pdf

[25] Alex Borges Vieira, Wallace Nascimento Paraizo, Luciano Jerez Chaves, Luiz Correia, and Edelberto Franco Silva. 2022-04. An SDN-based Energy-aware Traffic Management Mechanism. 77, 3 (2022-04), 139–150. https://doi.org/10.1007/s12243-021-00863-x

[26] Moshe Zukerman, Timothy D. Neame, and Ron Addie. 2003. Internet Traffic Modeling and Future Technology Implications. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)* (San Francisco, CA, USA), Vol. 1. IEEE, 587–596. https://doi.org/10.1109/INFCOM.2003.1208709