Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Betsegaw Lemma Amersho

Simulating energy-aware networks in large-scale distributed systems

Master's Thesis
Espoo, June 26, 2017

DRAFT! — May 22, 2017 — DRAFT!

Supervisors:	Professor Antti Ylä-Jääski, Aalto University
	Professor Pekka Perustieteilijä, University of Helsinki
Advisor:	Olli Ohjaaja M.Sc. (Tech.)

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF MASTER'S THESIS

Author:	Betsegaw Lemma Amersho		
Title:			
Simulating energy-aware networks in large-scale distributed systems			
Date:	June 26, 2017	**Pages:**	44
Major:	Data Communication Software	**Code:**	T-110
Supervisors:	Professor Martin Quinson Dr. Anne-Cécile Orgerie		
Advisor:	Olli Ohjaaja M.Sc. (Tech.)		

A dissertation or thesis is a document submitted in support of candidature for a degree or professional qualification presenting the author's research and findings. In some countries/universities, the word thesis or a cognate is used as part of a bachelor's or master's course, while dissertation is normally applied to a doctorate, whilst, in others, the reverse is true.

!FIXME **Abstract text goes here (and this is an example how to use fixme).** FIXME! Fixme is a command that helps you identify parts of your thesis that still require some work. When compiled in the custom `mydraft` mode, text parts tagged with fixmes are shown in bold and with fixme tags around them. When compiled in normal mode, the fixme-tagged text is shown normally (without special formatting). The draft mode also causes the "Draft" text to appear on the front page, alongside with the document compilation date. The custom `mydraft` mode is selected by the `mydraft` option given for the package `aalto-thesis`, near the top of the `thesis-example.tex` file.

The thesis example file (`thesis-example.tex`), all the chapter content files (`1introduction.tex` and so on), and the Aalto style file (`aalto-thesis.sty`) are commented with explanations on how the Aalto thesis works. The files also contain some examples on how to customize various details of the thesis layout, and of course the example text works as an example in itself. Please read the comments and the example text; that should get you well on your way!

Keywords:	ocean, sea, marine, ocean mammal, marine mammal, whales, cetaceans, dolphins, porpoises
Language:	English

Acknowledgements

I wish to thank all students who use LaTeX for formatting their theses, because theses formatted with LaTeX are just so nice.

Thank you, and keep up the good work!

Espoo, June 26, 2017

Betsegaw Lemma Amersho

Abbreviations and Acronyms

2k/4k/8k mode	COFDM operation modes
3GPP	3rd Generation Partnership Project
ESP	Encapsulating Security Payload; An IPsec security protocol
FLUTE	The File Delivery over Unidirectional Transport protocol
e.g.	for example (do not list here this kind of common acronymbs or abbreviations, but only those that are essential for understanding the content of your thesis.
note	Note also, that this list is not compulsory, and should be omitted if you have only few abbreviations

Contents

Chapter 1

Introduction

1.1 Context of the Study

1.2 Problem statement

Even though there is a growing concern about the energy consumption of large-scale network infrastructures, a search of the literature revealed few studies which address the issue of estimating the energy consumption of these network infrastructures. The existing proposed solutions being packet-level estimators are not scalable (in-terms of memory usage and speed) to be used in the domain of large-scale distributed networks such as cloud, grid, and peer-to-peer computing.

1.3 Aim and Scope

The purpose of this work is first to propose and to implement flow-level energy consumption model in SimGrid and then to show that the implemented model gives reasonably accurate result and it is also scalable for estimating energy consumption of large-scale distributed networks.

In the domain of large-scale distributed network infrastructure, we can categorize energy consuming components into two broad groups: IT equipments (which includes computing servers, storage servers and networking components) and infrastructure components (which includes power provisioning, cooling and lighting components). Since SimGrid already have energy consumption model for computing servers, the scope of this study is on the network component part within the IT equipments category.

1.4 Significance of the Study

1.5 Structure of the Thesis

Chapter 2

Background

In this chapter we first start by describing the current trend in global electricity consumption in the IT field and then we describe the energy consuming components involved in large scale distributed networks. In related to this we describe the concept of energy proportionality, which explain why servers and network devices are considered energy inefficient. We then mention the approach used to study this energy inefficiency problem. We gave particularly emphasize on packet-level and flow-level simulators. Next, we describe SimGrid as a large-scale distributed network simulator, its role in estimating energy consumption of large-scale distributed networks, and what needs to be done for its role. Finally, we review existing simulators that are proposed for estimating large-scale network energy consumptions.

2.1 Electricity consumption of ICT equipments

ICT devices consume a significant amount of electricity. A survey conducted by Heddeghem et al. [15] shows the electricity consumption and growth trends of three classes of ICT equipment: personal computers, communication networks, and data centers. Personal computers include equipment such as desktop, laptop and external monitors. Communication networks includes residential network access equipments (such as WiFi routers and modems), network equipments used in offices (such as routers and switches) and telecom-operator network equipments (such as base stations, routers and optical amplification systems). Data-centers house storage and computing servers, communication network equipments, and power provisioning and cooling facilities. In this classification there are overlaps, for instance, telcom operator can have office network equipments and data-centers. After carefully avoiding possible redundant measurements, the researchers estimated absolute electricity consumption and annual consumption growth rate of each category of equipments for the period 2007 and 2012. The results of the study show that the the global electricity consumption share of personal computers is 1.6%, communication networks is 1.7%, and data centers is 1.4%. The estimated annual growth rate of each category is 5% for personal computers, 10% for communication networks, and 4% for data-centers. These growth rates are higher than that of the total global electricity consumption, which is 3%. This trend signifies the need for energy saving research in all the three categories.

2.2 Data-center and network electricity consumption

In Section 2.1 we described data-center's global share in electricity consumption. In this section we describe the components involved within the data center itself.

Electricity consumption units within a typical data-center can be classified into two broad groups [11]: The first group is IT equipment (which includes computing servers, storage servers and networking components) and the other group is infrastructure facilities (which includes power provisioning, cooling and lighting components).

Figure 2.1 from [11] shows the electricity consumption proportion of the data-center components. This value differs significantly from one data-center to another [2], for instance, due to architectural difference[14] or energy efficiency of the components. The infrastructure facility components take the large proportion (65%) of the consumption. Though the infrastructure facility consumes relatively larger amount of electricity, the focus of this study is on the IT equipment components, particularly on the network equipment.

If we further zoom in on the IT equipment part, we can find computing servers, storage servers and network devices. A data-center servers consist of one or more CPU cores, memory and I/O devices. The energy consumption relationship among these components is shown in Figure 2.2. Combined, Memory and CPU units consume the larger amount of energy relative to other components. The fact that CPU is the dominant electricity consuming unit is exploited by Fan et al. in [12] to model the dynamic power usage of thousands of servers by using only CPU utilization as a parameter. The result of their study was very accurate, with error as low as 1%. The energy consumption contribution of storage servers in a typical data center is shown

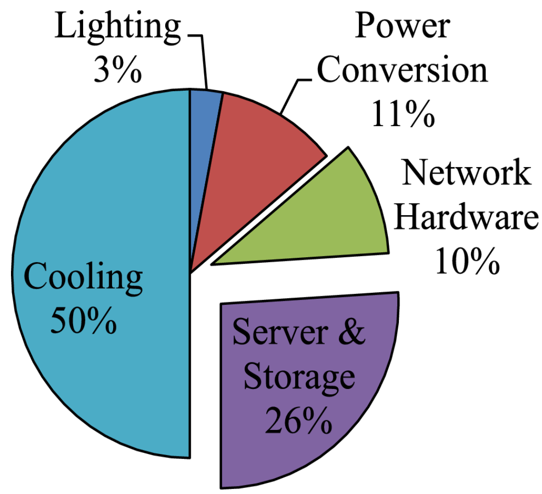

Figure 2.1: Energy consumption percentage of data-center components from [11]

in Figure 2.1 together with computing servers.

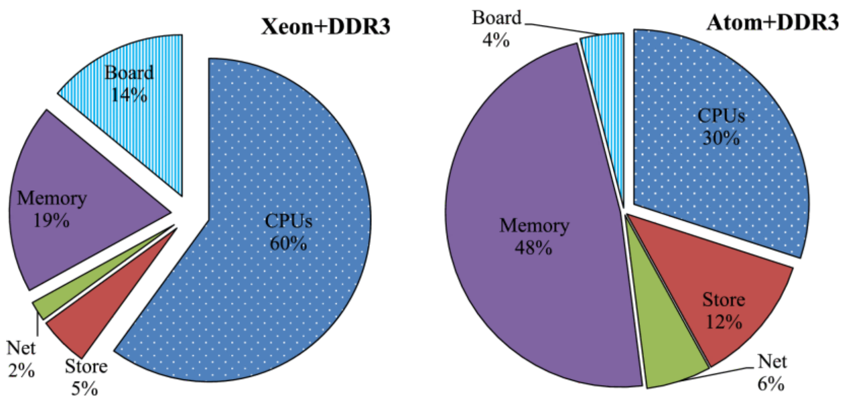

Figure 2.2: Energy consumption percentage of Xeon based (on the left) and Atom based (on the right) servers [11]

Network devices are the other part in the IT equipment component of a data center which contribute to energy consumption as shown in Figure 2.1. Shehabi et al. [23], from Berkeley National Laboratory, produced a report which show the annual energy consumption of network devices deployed in data centers residing in the United State. The historical and the forecast energy consumption is shown in Figure 2.3. In the figure the energy consumption is grouped by port speed of 100Mbps, 1000Mbps, 10Gbps, and 40Gbps.

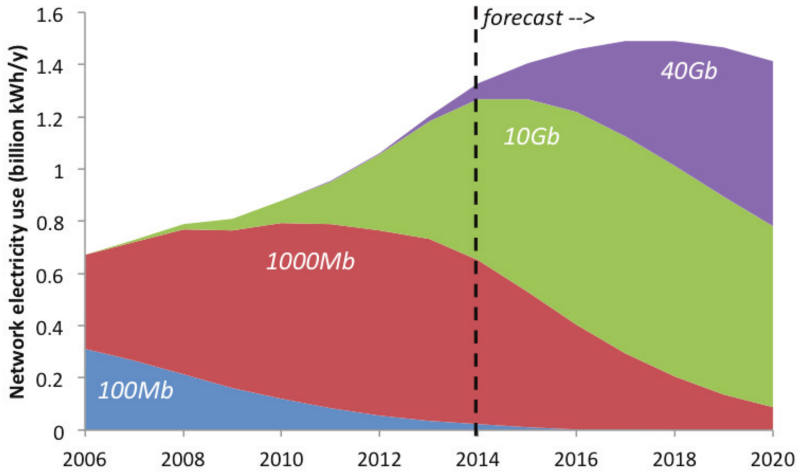

Figure 2.3: Total data center network equipment energy consumption in the United States [23].

In large-scale distributed networks, network devices are deployed with in and outside the data center. Our study is not limited only to network devices residing in a particular data center, it also includes network devices residing outside a data center.

2.3 Energy proportionality

The primary reason the study of energy consumption management of network equipment becomes so important is that, in general, ICT equipment do not consume energy proportional to their workload. An ideal ICT equipment is the one which consume zero electricity when it is idle, and it consumes electricity proportional to its workload when it is active. However, the reality is, even power efficient servers consume about 50% of their peak power [3], even when they are doing nothing. This percentage can even reach 85% for network switches [13]. Figure 2.4 from [20] shows the energy proportionality of a typical network equipment. From the graph we can observe that the dynamic power consumption range is narrow. Three approaches are in common use to deal with this situation. The first one is re-engineering network devices so as to make them more energy proportional, device vendors are

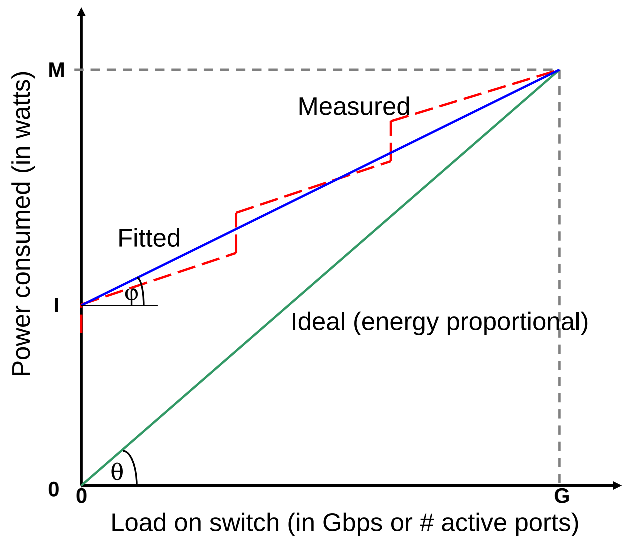

Figure 2.4: Ideal and measured energy proportionality of a network equipment [20]

the prime role player in this aspect. The second approach is related to the operating rate of a network equipment port. A typical switch can operate on different transmission rate (100Mbps, 1 Gbps or 10 Gbps). An active port transmitting at 10 Gbps can consume more energy than if it transmit at 100 Mbps. Rate adaptation is the approach devised to take advantage of this situation. Instead of transmitting at the maximum rate all time, the network port can be made to adapt to the actual traffic load. This energy saving approach is known as Adaptive Link Rate (ALR)[21]. The third approach, which is known as Low Power Idle (LPI), allows a network device to send data as fast as possible and then enter low power mode between transfers [3]. The low power mode can further be extended by a technique called packet coalescing, which allows more energy saving [5].

2.4 Packet-level and flow-level Simulators

One way of conducting experiment is to use real production environment or test-bed environment, both are referred to as *in vivo* in [8]. In the former case, handling transient and varying conditions would make the data collection and prediction very difficult and often times, a production environment is also not available for experimentation. In the later case, it requires setting-up a separate testing environment designed solely for the purpose of conducting the desired experiment. This approach apart from being expensive, it requires significant amount of time for experiment setup and, it is also non-repeatable as experimenting with different scenario demands a significantly modified or completely new configuration.

The other alternative for experimenting is simulation, also referred to as *in silico* in [8]. Simulation, unlike real environment, allows great flexibility in terms of experiment configuration, control and repetition. In addition it can also be less time consuming and less expensive.That is why virtually in all computer network related researches simulations are widely used.

Simulators use models to specify the relationship between the variables involved in a particular network phenomenon. Generally, the models are classified as packet-level and flow-level models based on the detail of information the models are trying to capture. We can also refer to simulators as packet-level simulator and flow-level simulators based on the model they use.

Packet-level simulators strives to model a given network phenomenon at the granularity level of individual packets. Due to the detail of information this simulators capture, in general, they are accepted by the research community to be more accurate compared to flow-level ones [8]. One of the most popular packet-level simulator is NS-3, which is categorized under discrete-event simulator with events corresponding to sending and receiving of packets [18]. Though packet-level simulators are accepted to be more accurate, they fail to scale well in the field of large-scale distributed networks due to the computation and storage cost involved in processing and storing each packet.

In the area of large-scale networks, flow-level simulators are the preferred simulation alternative. Rather than modeling a given network phenomenon at an individual packet level, flow-level models treat a set of packets as a single unit [8]. The most commonly used definition for flow in the context of computer networking is coined by Claffy et al. in [9]:

"...a flow ...a unidirectional traffic stream with a unique [source-IP-address, source-port, destination-IP-address, destination-port, IP-protocol] tuple ..."

In addition to the five tuple mentioned in the definition, a flow also has a limited time duration. Claffy et al. [9] used a time limit of 64 seconds as a flow duration in their study. Researchers such as Carneiro et al. [7], adopted this same definition to develop flow monitoring module for NS-3, a module that can generate information such as amount of packets or bytes transferred, packets dropped or transmission start and end time for each flow. Barakat et al. in [1] also used the same definition to model traffic at the flow-level for the Internet backbone link. By abstracting away fine details, flow-level models provides easy way to instantiate experiments and they also scale very well for conducting large-scale network simulations [1, 8].

The flow definition given above is not the only one. Any analytical model which capture the characteristics of a given network phenomenon can be considered as flow-level model. In SimGrid, for instance, TCP flow is characterized by bandwidth and end-to-end latency [8].

2.5 Simulating and modeling energy consumption of large-scale networks

In this study we simulate energy-aware large scale distributed networks using SimGrid (Detail description about SimGrid follows in the next section). When we say large-scale distributed network, we are referring to a set of networks residing inside in the distributed data centers and also the networks that are used to connect them.

The energy consumption E of an equipment depends on the operating power P at time t. The total energy consumption for a time period T is given by Equation 2.1 [22].

$$E(T) = \int_0^T P(t)dt \tag{2.1}$$

Due to the energy proportionality characteristic described in Section 2.3, the common approach used to compute the energy consumption is to divide the power component into two parts: static/idle power (P_{static}) and dynamic power ($P_{dynamic}$) as shown in equation 2.2. Then the total energy is obtained by multiplying the total power, P_{total} by the time duration [11, 17, 20, 22].

$$P_{total} = P_{static} + P_{dynamic} \tag{2.2}$$

For a typical network equipment such as a switch, the static part constitutes the power consumption of the chassis and the line-cards (when all the ports on the line-cards are switched off). The dynamic part, on the other hand, constitutes the power consumption of the switch ports running at a given rate multiplied by the utilization factor [20]. Equation 2.3 shows how to compute the total power for a switch, where P_{switch}, is the total power consumption of a switch, $P_{chassis}$ and $P_{linecard}$ is the idle power consumption of the chassis and the line card, respectively. P_{rate}, is the power consumption of a given port at a given rate and $numports_{rate}$ is the number of ports running at a given rate. The rate can take values such as 10 Mbps, 100 Mbps, 1 Gbps or

10 Gbps.

$$P_{switch} = P_{chassis} + (numlinecards \times P_{linecard}) + \sum_{rate=min}^{max} (numports_{rate} \times P_{rate} \times utilizationFactor) \tag{2.3}$$

2.6 SimGrid

SimGrid is one of the popular simulator available for simulating large-scale distributed networks such as grid, cloud, volunteer and HPC [27]. It employs flow-level models in its core for simulating different network resources and phenomenon. In subsequent paragraphs we give overview of its architecture, the pros and cons of the employed TCP flow-level model and its current status in relation to energy consumption simulation models.

Figure 2.5 shows the structure of SimGrid and how its core works. The top three components are the APIs that users can use to develop their simulation. Both MSG and SMPI are used to specify simulated applications as concurrent processes. The difference is that using MSG, users can simulate any arbitrary application, whereas, using SMPI users can simulate existing MPI applications, the MPI processes are created automatically from C or Fortran MPI programs. SIMDAG, on the other hand, does not use concurrent processes. It allows users to describe their application as communicating task graph. The next layer, SIMIX, implements the mechanisms that are required to simulate the concurrent process of MSG and SMPI applications. It also provides process control and synchronization functionalities. The bottom layer, SURF, is the simulation core, it simulates the execution of activities on computing or communication resources [8]. In SimGrid for each simulated activity, such as computation or data transfer, there is a corresponding condition variable, in Figure 2.5 it is shown in SIMIX box. This condition variable synchronizes the concurrent processes of the simulated applications. The computing (P_x) and the communication (L_x) resources are shown on the bottom-right side of the figure. Computing resources are defined in terms of computing power, whereas, communication resources are defined in terms of bandwidth and latency. As shown in the SURF box, multiple activities can share the same resource (e.g., (x_1, x_n), (x_1, x_3) or (x_3, x_n)) or one activity can use multiple resources (e.g., x_1 or x_3 or x_n). Activities that share the same resource are limited by the capacity of that resource. Each activity is defined by the total and remaining work to be executed. When the work associated with the activity completes, the corresponding upper layer components receive a notification signal [8].

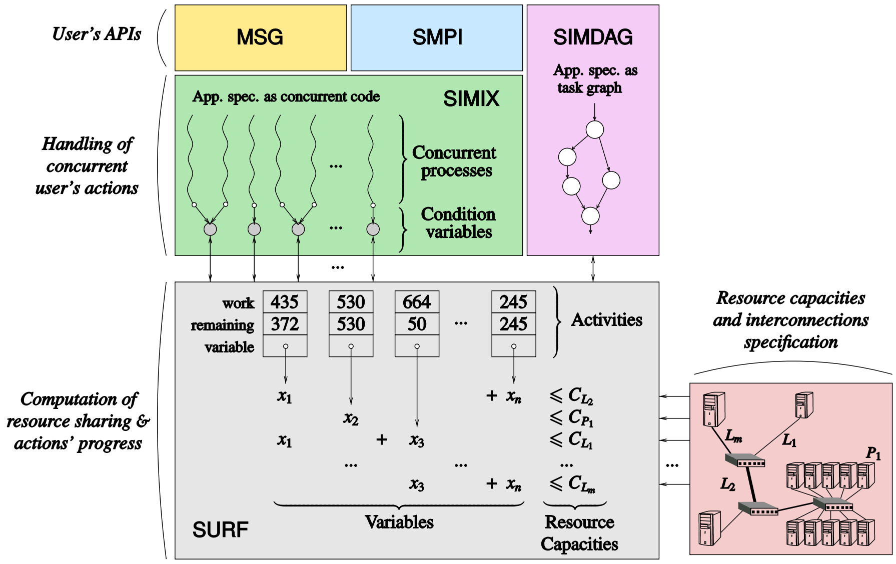

Figure 2.5: Architecture of SimGrid [8]

As we have already pointed out in Section 2.4, the primary advantage of flow-level simulation is its scalability in terms of speed and memory usage. SimGrid uses flow-level analytical model for simulating TCP network phenomenon [8]. To show the scalability of the flow-level model, the SimGrid team compared it with other widely used simulators such as GridSim and OverSim. After simulating 500,000 tasks both on GridSim and SimGrid, the results demonstrate that SimGrid is 257 times faster and 26 times more memory efficient. Similarly, the comparison result with OverSim shows that SimGrid is 15 times faster and it can also simulate scenarios 10 times larger. Concerning the accuracy, though the simulator gives very good accuracy in most case studies, there are situations where it fails to give accurate result. As an example, the comparison study of SimGrid with packet-level simulator GTNetS shows that for data size less than 100 KiB there is a significant difference in prediction.

Currently, SimGrid has energy consumption model for CPU. Using this CPU model researchers can simulate energy consumption of single or multi core CPUs running at different operating frequencies. Concerning network equipment however, SimGrid has no energy consumption model. Therefore, the focus of this study is to propose and implement network energy consumption model for SimGrid. The implementation of this model, together with the existing CPU energy model, allows us to estimate the energy consumption of large-scale networks that reside within or outside a data-center such as networks discussed in Section 2.1 and Section 2.2.

2.7 Related Simulators

In this section we review existing simulators that are proposed for estimating energy consumption of large-scale networks.

2.7.0.1 ECOFEN

Orgerie et al. [22] proposed ECOFEN, an Energy Consumption mOdel For End-to-end Networks. It is a packet-level simulator designed for estimating energy consumption of large-scale networks. Initially the simulator was developed as NS-2 module but currently it is also available as NS-3 module [10]. ECOFEN provides three models for simulating energy consumption at different levels of granularity: *basic*, *linear* and *complete*.

The basic model allows to simulate energy consumption of a network interface card (NIC) at coarse level of granularity. It only accept energy consumption value for ON and OFF state of the NIC. The linear model, on the other hand, accepts energy consumption value for the idle state of the NIC and for each bytes processed. This model allows to compute power consumption of a given network traffic. The complete model like the linear model also considers traffic in its power consumption computation. The difference is that it offers added flexibility in terms of parameters. Different energy consumption values can be assigned for bytes received or send and also packets received or send. All the three models produce the estimated average energy consumption at milliwatt precision level in the chosen time interval and the time interval can be set as small as a millisecond.

ECOFEN module has two main limitations, mainly due to the limitation of the underlying NS-3 simulator. The first limitation comes from the lack of CPU abstraction in NS-3. As we have discussed in Section 2.2, server energy consumption is the second dominant part in typical data center and CPU is the main contributer among server parts such as memory and storage. Furthermore, since the energy consumption of CPU is linearly dependent on its operating frequency and its workload, the energy consumption increases more as the workload increases. As a consequence of absence of CPU from NS-3, the energy estimation we get from ECOFEN is partial. It only able to simulate energy consumption of network components such as NIC, switches and routers. The second limitation is concerned with the scalability issue. Being a packet-level simulator, the performance of ECOFEN is affected significantly as the number of processed packets grows large. Cornea et al. [10]

noticed this scalability problem during their study of the energy consumption of data transfers in clouds using ECOFEN module. It took them 5 hours to capture 1 minute of simulated network activity for a large-scale network.

2.7.0.2 GreenCloud

Kliazovich et al. [17] proposed GreenCloud, a simulator that can estimate energy consumption of cloud computing data centers. GreenCloud is developed as extension to NS-2 packet-level network simulator. This simulator contains power consumption models both for the computing and communicating components of a typical data center. The power consumption model used for the computing component is shown in Equation 2.4. This equation contains power consumed by the fixed parts (such as bus, memory and disk) which consume power independent of the operating frequency f of the computing component CPU and the power consumed by the CPU (P_f) operating at a given frequency f. This model allows for lowering the operating frequency of the CPU when workload becomes below some predefined threshold in order to decrease the power consumption.

$$P_{computing} = P_{fixed} + P_f \times f^3 \tag{2.4}$$

h= The power consumption model used in GreenCloud for the communicating components is the one shown in Equation 2.3. The equation shows the static power consuming parts (such as the chassis($P_{chassis}$)and the active line cards($P_{linecard}$) and the dynamic part (P_{rate}) is the energy consumed by the port running at a particular line rate for a given traffic load.

This simulator is limited in four aspects: (1) in the number of allowed CPU cores, (2) in versatility, and (3) in scalability. The first limitation is that only one CPU core is allowed per simulated node. This hinders the study of energy consumption of multi-core computing nodes. The second one is that we can not use this simulator outside the cloud computing domain such as grid, volunteer, peer-to-peer or HPC, at least that is not the authors original intention when they develop this simulator. The available features of the simulators are tuned towards cloud computing applications only. This limits its versatility. The third limitation deals with the scalability issue. The fine grain details provided by GreenCloud and the packet-level processing approach of the underlying NS-2 simulator is advantageous for getting accurate result when simulating relatively small networks. However, for large-scale distributed networks, it is not scalable. In related to this, the authors have mentioned that the simulation speed gets slower and slower as the number of simulated nodes increases beyond few thousands and also as the number of processed packets increase. The GreenCloud's underlying simulator NS-2, is known for its scalability problem. Currently NS-3 is available as a better performing alternative [29] however, we could not find any upgraded for GreenCloud.

Chapter 3

Environment

In this study we employed SimGrid, ECOFEN and FlowMonitor modules of NS-3 simulator. This chapter explains the main features of these tools from the perspective of our simulation experiment needs.

3.1 SimGrid

In Section 2.6 of Chapter 2 we discussed the software architecture of SimGrid at a higher level. We will give low-level details of the implemented flow-level model and related concepts in later chapter. In this section, our plan is to discuss features of SimGrid that are related to setting up and running energy consumption experiments.

TODO: Include how SimGrid experiments are done

3.2 NS-3

NS-3 is a discrete-event packet-level simulator, events corresponding to, for instance, arrival and departure of packets. NS-3 is structured in a modular manner. The core and the network modules are two of the modules that serve as generic simulation core that can be used for Internet-based or different network type simulation. These two modules, being generic, are independent from any device models. The core module provides features such as tracing, callbacks, smart pointer, random variables, events and schedules. The network module consists components such as packets, node, addresses (e.g. IPv4 and MAC) and network devices. The components provided by the simulation core modules can be used to create other modules. This feature allows researchers to add their own models for the network phenomenon that they want to simulate. We will visit two of the modules that are constructed in this way in the next two subsections[18].

The NS-3 core and other modules are built in C++ language as a set of libraries. The user can access these libraries in their main C++ program to configure the simulated topology and other simulator parameters. The libraries are also available as Python API for Python programmers.

3.2.1 ECOFEN Module

ECOFEN is one of the two non-core NS-3 modules that we used in our experiments. We explained the power consumption simulation features provided by this module in the Related Simulators section of Chapter 2 and we also give detailed explanation about why and where we have used it in our Method chapter. In this section, we only give brief description about how it is related to NS-3 and how we have used it.

NS-3 in its core provides an abstraction such as Node, Net Device, Channel and Application. A Node represents network communication and computing devices such as servers, switches and routers. To a Node a Net Device, which represent devices such as network interface card(NIC), can be attached. Two or more Nodes can be linked to each other through a Channel, which is a representation of Ethernet or Wi-Fi link. These three abstractions: Node, Net Device, and Channel, together they can be used to define the simulated network topology. Application, on the other hand, is an abstraction that represent user program that perform some simulated activity such as sending or receiving UDP packets for instance[18].

Using the core abstractions provided by NS-3, such as Node, Net Device, and Packet, the ECOFEN module implemented three power consumption models that enable users to simulate power consumption in relation to packets transferred at different levels of granularity as discussed in Chapter 2 and Chapter 4.

In a typical NS-3 simulator script, we can recognize four common sections in the main function: (1) the section where we find statements that import the required core or other modules, (2) the section where the topology of the simulated network is defined, (3) the section where the simulated user application is defined, and (4) the section where statements related to running, starting, stopping and cleaning the simulation is specified. This is a rough approximation, certainly there are other statements such as those which are related to logging and tracing.

The NS-3 scripts that we used in our power consumption simulation experiments imported the ECOFEN module in their first section, set up and configured the energy consumption models in the second section, and the remaining sections are as described above.

3.2.2 FlowMonitor Module

FlowMonitor is the other non-core NS-3 module that we have employed in our study. This module is designed with the aim of providing generic network traffic inspection facility for researchers who want to measure the simulated network efficiency using standard performance metrics such as bit-rate, duration, delay, packet-size and packet loss ratio[7].

Among the performance metrics that are available in FlowMonitor module, the following are the ones we have used in our simulation experiments.

- ***rxBytes*** to get the received bytes by a node,
- ***txPackets*** to get the transmitted packets by a node,
- ***timeFirstRxPacket*** and ***timeLastRxPacket*** to get the absolute time when the first packet and the last packets in the flow was received,
- ***timeFirstTxPacket*** and ***timeLastTxPacket*** to get the absolute time when the first and last packets in the flow was transferred and
- ***lostPackets*** to check if there are lost packets.

We use the above performance metrics to compute throughput (T) with the unit of Mega-bits per second (Mbps) and Packets per second(Pps) as shown in Equation 3.1 and Equation 3.2.

$$\begin{aligned} T_{Mbps} = {} & rxBytes \times 8.0 \times 10^6 / \\ & (timeLastRxPacket - timeFirstRxPacket) \end{aligned} \tag{3.1}$$

$$T_{Pps} = txPackets/(timeLastTxPacket - timeFirstTxPacket) \tag{3.2}$$

Chapter 4

Methods

In this chapter we begin by first describing the common approaches followed by researchers for a variety of energy consumption experiments, their advantages and disadvantages. Then we present the approach we followed in our study and its justification.

4.1 Common Approaches

One approach for estimating energy consumption of a given network is by employing actual power meter to measure the power drawn by involved network and computing components. A good example for such case is the measurement that Fan and his team conducted[12]. In this study the authors have managed to monitor power consumption of several thousand of servers over a period of six months on real live workload. Mahadevan et al. in [20] have also done a similar power measurement on a production environment for studying power consumption behavior of networking devices such as switches and routers. If the measurements are done correctly, this approach produces the most real picture of the network under investigation compared to the other approaches discussed in subsequent paragraphs. However, this approach has certain inherent drawbacks. First, real production networks might not be available for experimentation. Even if they become available, the transient and varying nature of the production environment makes it hard to repeat the experiment. Second we have little or no control over factors affecting the measured power consumption. We do not have the privilege of injecting or modifying the traffic or the workload in order to test different experimental hypothesis. To have a full control we need another approach. That is what we discuss next.

Experimental testbed is another approach that researchers have used to study power consumption characteristics of different computing and networking devices. In this approach first a separate network is setup and configured solely for the purpose of conducting experiments. Then researchers make measurements by manipulating factors that affect power consumption according to the hypothesis that they want to test. Unlike the previous one, this approach offers greater flexibility over the experimental parameters. In the power measurement study scenario that we are discussing, the researcher can change parameters such as traffic rate, packet size, inter-packet time interval and transmission protocol used (TCP/UDP). Sivaraman et al. in [25] have setup experimental testbed for determining per-packet processing and per-byte receipt, storage, queuing, and transmission power consumption. The experiment setup involved hardware-based traffic generator (which gives fine grain control over parameters such as the packet size, inter-packet interval and data rate), NetFPGA[1] experimental router and digital oscilloscope for measuring the power draw of the NetFPGA router. A similar experiment but with commercial switches of different vendors is explained in [24]. The primary advantages of this approach is that the researcher can have full control over the experimental parameters provided by the tools involved in the testbed and experimental result can also be very accurate. The first disadvantage though is that it can easily become very expensive when we want to experiment on large-scale level. The second disadvantage is that experimenting on different scenario might require considerable reconfiguration and even a completely new testbed, which apart from limiting the flexibility, it can also be very costly, time and effort consuming. We need an approach which overcome these shortcomings. That is, we need an approach which gives full control over the experiment, which is reasonably accurate, less expensive and very flexible.

Simulation is the most widely used approach in computer network researches [29]. It has several advantage compared to the other two approaches mentioned before. First, it makes it relatively easy, for instance, the study of the performance of non-existing network protocol or algorithm. One can propose and validate, by simulation experiment, a new energy-aware routing protocol or algorithm for wired or wireless networks. This is what Swain et al. [26] did in their new energy-aware routing protocol proposal for wireless sensor networks. Second, though it depend on the design of the particular simulator used, in general, simulation approach allows running large scale experiments that involve hundreds and thousands of nodes with less effort and cost compared to the other two approaches. In [22] and [10], the NS-3 module, ECOFEN, is used to simulate energy consumption of large-scale

[1]http://www.netfpga.org/

networks with nodes more than 600 and 1000, respectively. In [17], the authors studied energy consumption of data center networks with two-tier and three-tire architectures that encompasses 1536 nodes. Third, in simulation scaling does not incur monetary cost, rather, it is limited by performance factors such as runtime and memory usage [29]. Fourth, the researcher has great flexibility and full control over the simulation experiment. Finally, simulators makes event output data management extremely easy by providing mechanisms such as logging, tracing and visualization [8, 18].

Though simulation experiment has quite a lot of advantages over experiments done on production environment or experimental testbeds, it faces one big challenge, accuracy. In the process of approximating the real network phenomenon in the simulation model, some less significant concepts are abstracted away, for instance, to reduce complexity or to gain performance improvement, which results in unavoidable loss of accuracy. However, in other instances the models used in a given simulator might fail to correctly capture the simulated real network phenomenon. In [28] the authors demonstrated incorrect modelings found in popular simulators such as OptorSim, GridSim and CloudSim. Therefore, (in)validating the correctness of a simulator is important task that should be undertaken before any simulation experiment for two related reasons. Either to know the boundaries within which the simulator used produce reasonably accurate results, or to know if the simulator produce the expected or the correct result. The validation can be done either by comparing the output of the simulator against accurate measurements obtained from real networks or by comparing the output against another simulator whose accuracy is already known [16].

4.2 Our Approach

The purpose of this study is to implement analytical or flow-level (as opposed to packet-level) energy consumption model for SimGrid and to show that the implemented model produce reasonably accurate result and is also scalable for estimating energy consumption of large-scale distributed networks. To achieve this goal, we use the simulation approach among the three alternatives discussed above.

Before describing the details of our approach, let us first justify why we end up with the relatively complex validation methodology shown in Figure 4.1. There is experimental test-bed (Grid'5000[2]) in France that we have access to. Grid'5000 is experimental test-bed specifically designed for

[2]https://www.grid5000.fr/mediawiki/index.php/Grid5000:Home

studying large-scale distributed networks [6]. However, we could not used it for our purpose (i.e., for studying large-scale flow-level relationship of power consumption and traffic) as the network devices are not equipped with power meters accurate enough (current power meters on Lyon site of Grid'5000 provide one measurement per node and per second). As a result, we opted to use a packet-level simulator with power consumption models obtained from literature. Subsequent paragraphs describe the specific steps we followed in our approach.

As we have discussed in Chapter 2, SimGrid already have energy consumption model for CPU which corresponds to the computing part of a given large-scale network. What we wanted to add is energy consumption model for communication components such as switches and routers. Therefore, the initial task in our approach is to study literatures (A in Figure 4.1) in order to find a model which describe the power consumption characteristics of communication equipments such as switches and routers. Our search returned the linear relationship that we have described in Equation 2.2 [4, 19, 20, 25]. This equation tells us that the power consumption of a network equipment constitutes the idle and dynamic components. The idle power consumption represents the power drawn by the equipment while it is on but with no traffic. The dynamic consumption, on the other hand, represent the additional power drawn due to network traffic. The next task (C in Figure 4.1) is to implement this linear model for SimGrid and (in)validate its accuracy against ECOFEN module (D in Figure 4.1)[10, 22]. The final task (G in Figure 4.1) is to show the scalability of the implemented flow-level model against the existing packet-level model in ECOFEN. For this we design and run two kinds of experiments (E and F in Figure 4.1), one for speed and one for memory usage.

We chose to use ECOFEN as packet-level simulator to compare the accuracy and performance of the implemented model for two primary limitations apparent in the other alternative simulator, GreenCloud [17]. The first limitation is that GreenCloud is designed for cloud computing environment. This is in contrary to one of SimGrid's main designed principle, versatility [8]. ECOFEN, on the other hand, is not tied to one particular large-scale networking paradigm, therefore, suits more for our purpose. The second limitation of GreenCloud is that it is build on top of currently obsolete NS-2 simulator. In comparison, though ECOFEN was also initially built as NS-2 simulator module, currently it is rewritten for NS-3 [10]. One of the major advantage of using NS-3 over NS-2 is that NS-3 performs considerably better in both runtime and memory-usage metrics [29].

In the accuracy-validation and scalability-comparison experiments mentioned in our approach, we are comparing the newly implemented flow-level model in SimGrid simulator against another packet-level simulator model implemented in ECOFEN module. This simulator-to-simulator comparison is valid only if the later simulator model, against which the new implementation is to be validated, is known to be accurate. However, we could not find any information that tell us the accuracy of the ECOFEN module. Therefore, we designed a validation experiment (B in Figure 4.1) for ECOFEN as described in the next section.

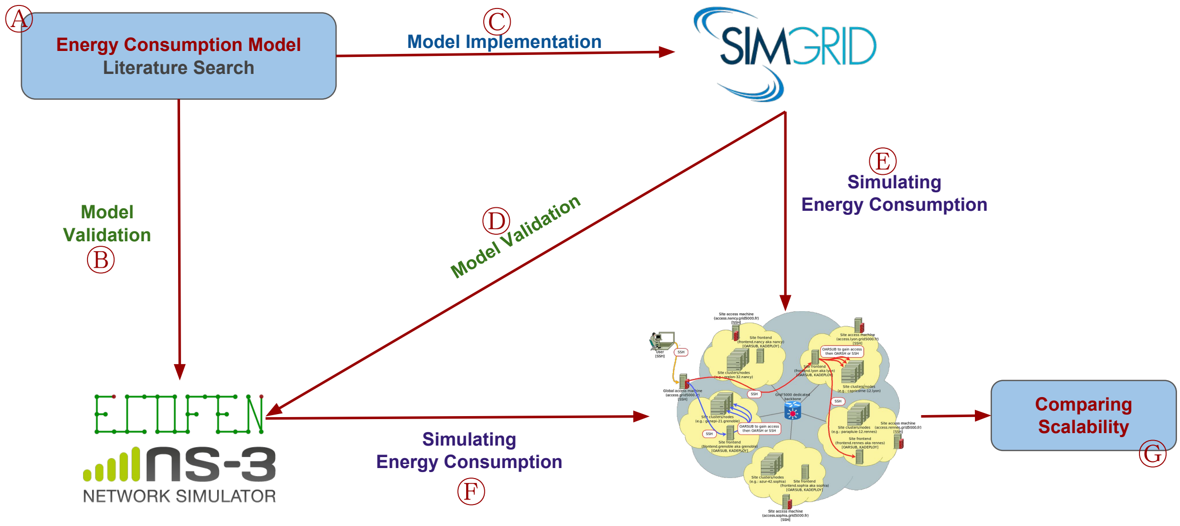

Figure 4.1: Summary of the methodology we followed in this study

4.3 Validating ECOFEN

ECOFEN has three models with names *basic*, *linear* and *complete* that we have discussed in Section 2.7 of Chapter 2. Both the *linear* and *complete* models can produce values of power consumption as a function of traffic. The *basic* model, on the other hand, produces power consumption values based on the ON or OFF state of a node, it does not consider network traffic. Therefore we describe the validation experiments for both the *linear* and the *complete* models in this section.

The basic procedure for the validation experiment is first to simulate, using ECOFEN, power consumption in response to traffic sent or received and then to compare the results against data obtained from actual measurements.

4.3.0.1 Validating the Linear Model

In the work of Sivaraman et al. [25] we can find the result of a power consumption experiment that is shown in Figure 4.2. The figure displays the linear relationship that exist between traffic volume (in Mbps) and power consumption (in watts) for a fixed packet sizes of 100, 576, 1000, and 1500 bytes. Furthermore, in the figure, the linear fit equations (models) for each of the packet sizes are also displayed.

The authors intention in this experiment is to determine values of the per-byte and the per-packet processing energy consumption, however, our intention is to use the per-byte energy consumption value that they have experimentally determined and to use it in ECOFEN to get power consumption values for a given volume of traffic. Then compare the results we obtained with the actual power measurement values shown in Figure 4.2.

In their experiment, the authors used three kinds of hardware devices: (1) NetFPGA router card that has four 1 Gbps Ethernet ports, (2) IXIA hardware traffic-generator for generating packets with the desired packet-size and data-rate, and (3) high-fidelity oscilloscope for measuring the power consumed by the NetFPGA card as a consequence of the packets send or received.

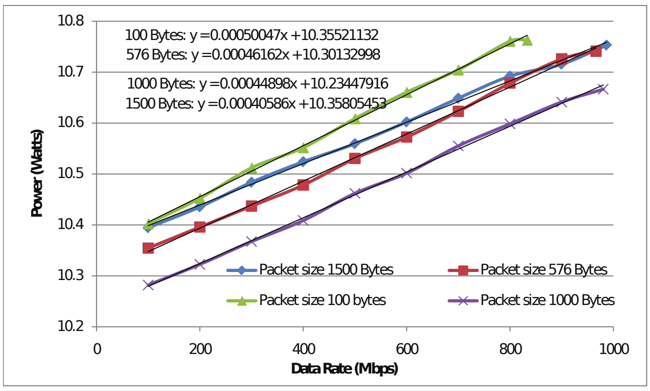

Figure 4.2: Power consumption vs data-rate for fixed packet size from [25].

The linear model of ECOFEN module accepts energy consumption values for the idle state of a simulated network interface card(NIC) and the per-byte processing. The underlying NS-3 platform, in addition, provides us with more parameters such as packet-size and data-rate, which among other parameters, enable us to have full control over the generated traffic.

In this validation experiment we wish to simulate the experiment conducted by Sivaraman et al. as closely as possible. With this in mind, we setup, in our NS-3 simulation script, a three node simple network with first and third nodes connected to the second node. All the three nodes are connected to each other by links that have maximum bandwidth capacity of 1Gbps and delay of 10ms.

We got the idle consumption values for each of the packet-size models shown in Figure 4.2 by setting the x component (the data rate value) to zero and for a per-byte processing energy consumption value we used 3.4nJ. This is the value that the researchers experimentally determined.

For the generated traffic volume in the simulation, we used uniform random number generator provided by NS-3 in order to get integer values between 1 and 1000. Our NS-3 script, in addition to packet-size and data-rate values, also requires number of packets to be send and also inter-packet interval time values. These values are derived from packet-size and data-rate parameters.

Since there might be unexpected results, for instance, due to wrong network configuration, we have employed NS-3's FlowMonitor module to monitor the actual traffic transfered in the simulated network. Using this flow monitoring module, we have confirmed if all the traffic generated by the sending end are received at the receiving end. We have also used this module to compute the actual traffic rate (throughput) both at the sending and the receiving ends.

Finally, we set the remaining simulation environment configuration settings such as starting and stopping time and then run the experiment 40 times, each time with different run value for the random number generator. The result obtained is depicted in Figure 4.3. In the graph the expected power consumption values from the linear fit models shown in Figure 4.2 along with the simulated values for each of the packet sizes (100, 576, 1000, and 1500 bytes) is displayed.

Visually, the simulated and the expected values seems to agree very well, even though the gap between them starts to grow slightly larger (especially when the packet size is 100bytes) for larger data-rates. In order to be more sure, we run unpaired t-test statistical test using the produced data. The summary of this test is shown in Table 4.1.

The 95% confidence interval values shown in Table 4.1 of difference in mean between the measured and simulated values are very close to zero and in fact zero is also one of the values. The P-values are also confirming the same thing, the null hypothesis that the difference in mean between the simulated and the expected values is zero is not rejected.

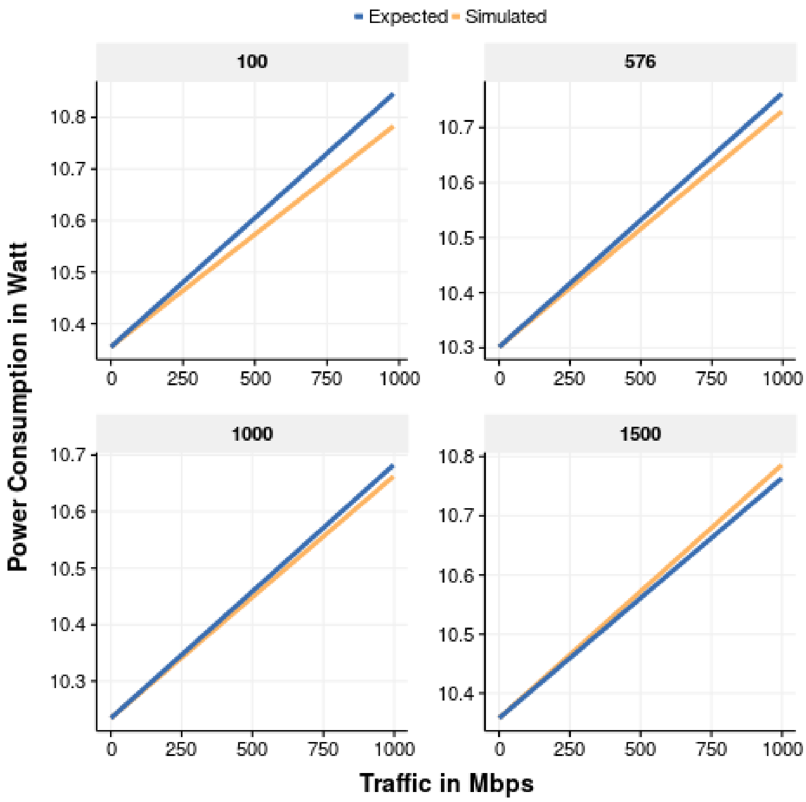

Figure 4.3: Power consumption vs data-rate comparison between expected (or measured) values (in red color) and simulated values (in light blue color) for a fixed packet size of 100, 576, 1000, and 1500 Bytes

.

The conclusion in this validation test is that the linear model of ECOFEN is accurate in predicting the power consumed by NetFPGA router for a given volume of traffic.

4.3.0.2 Validating the Complete Model

Roughly, in this validation experiment we have used the same experimental configuration and procedure as that of the linear validation experiment that we have described in the previous section. Therefore, in this section our focus will be more on the result than on the configuration.

Packet Size	Confidence Interval of difference in mean	Mean of Expected	Mean of Simulated	P-Value
100	[-0.027, 0.110]	10.640	10.599	0.230
576	[-0.039, 0.082]	10.544	10.523	0.480
1000	[-0.043, 0.073]	10.466	10.451	0.6131
1500	[-0.062, 0.048]	10.566	10.573	0.796

Table 4.1: Unpaired t-test results for simulated and measured power consumption values for ECOFEN's linear

One of the main difference between the complete and the linear model of ECOFEN is that the complete model distinguishes between the received and sent bytes. Which means that different energy consumption values can be assigned to bytes based on the direction of transfer. The linear model, on the other hand, assigns same value for both. The other main difference is that the complete model considers the packet processing energy consumption cost both for the sent and received packets.

Sivaraman et al. [25] also conducted experiments to determine energy consumption values for per-byte receive or transmit and per-packet processing. The experimentally determined values for per-byte receive is 1.3nJ, for per-byte transmit is 2.1nJ, and for per-packet processing is 197.2nJ.

We have slightly modified the NS-3 script that we have used in the previous section to make it suitable for this experiment. Now we have configured our script for the ECOFEN's complete model to use energy consumption values for per-byte receive or send and per-packet processing. Further more, we have upgraded the link capacity between the nodes from 1Gbps to 2Gbps. Finally, we set the traffic rate in terms of packets per second in the sending end and in terms of Mbps in the receiving end in order to comply with the experiments done by mentioned authors. The results for the sending and receiving are is shown in Figure 4.4 and Figure 4.5 respectively. The linear fit models we have used for this validation experiment are also available in [25]. There is only one linear fit model (for packet size 1000 bytes) for the sending end and there are three for the receiving end.

Table 4.2 shows the unpaired t-test result for one packet-sizes in the transmitting side (Tx) and for two packet-sizes in the receiving side (Rx).

Again in this case the 95% confidence interval values shown in Table 4.2 of difference in mean between the measured and simulated values are very close to zero and in fact zero is also one of the values. The P-values are also confirming the same thing, the null hypothesis that the difference in mean

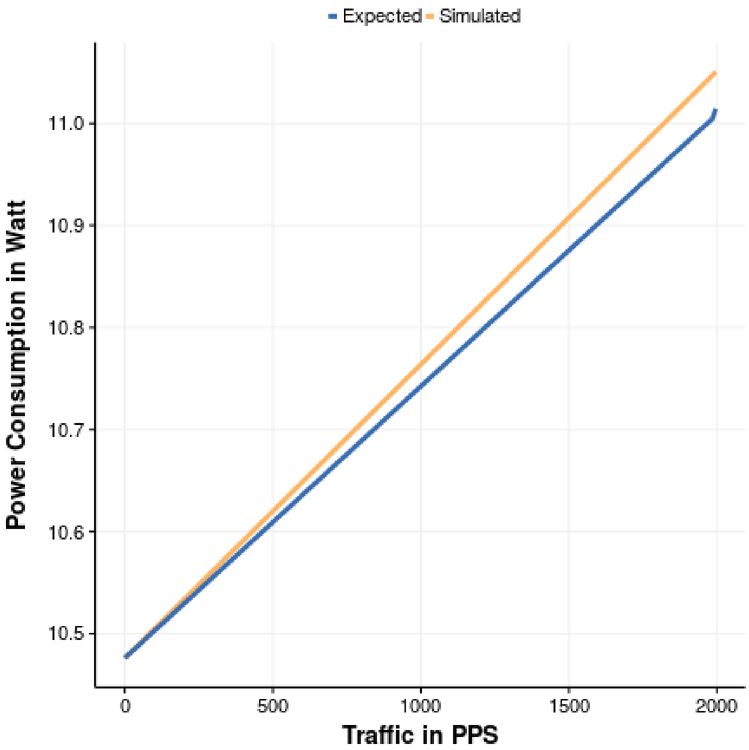

Figure 4.4: Power consumption vs data-rate comparison between expected (or measured) values (in red color) and simulated values (in light blue color) for a fixed packet size of 1000 Bytes for the sending end

between the measured and the simulated values is zero is not rejected.

The conclusion from this validation experiment is also the same as the previous one, the ECOFEN's complete energy consumption model accurately predicts power consumed by NetFPGA router for a given volume of sent or received traffic.

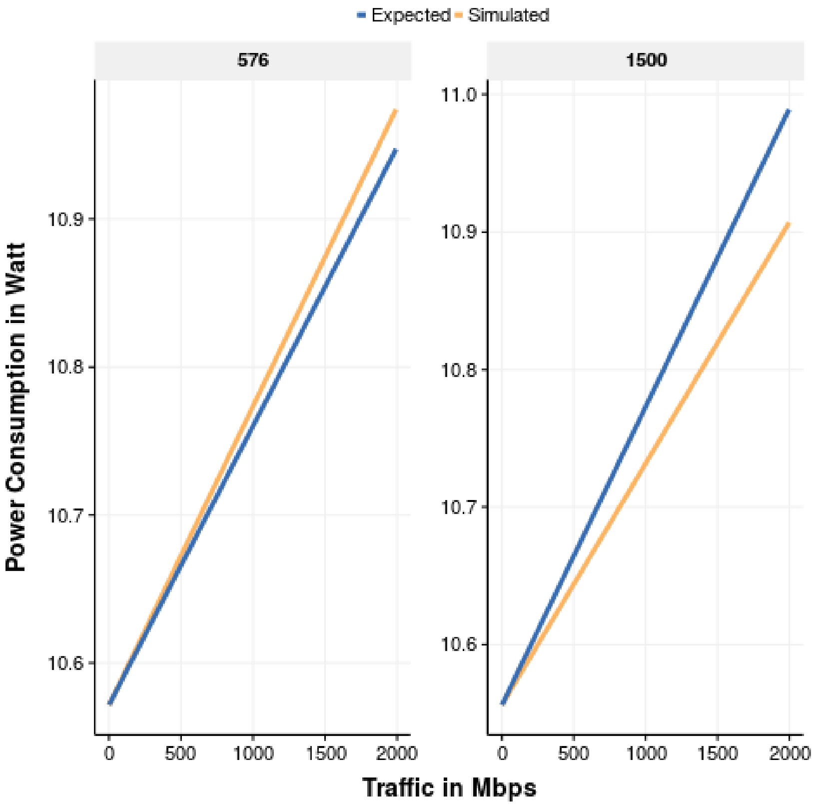

Figure 4.5: Power consumption vs data-rate comparison between expected (or measured) values (in red color) and simulated values (in light blue color) for a fixed packet size of 576 and 1500 Bytes for the receiving end

Packet Size	End	**Confidence Interval of difference in mean**	**Mean of Expected**	**Mean of Simulated**	**P-Value**
576	Rx	[-0.067, 0.039]	10.770	10.784	0.598
1500	Rx	[-0.010, 0.095]	10.778	10.736	0.114
1000	Tx	[-0.096, 0.053]	10.750	10.773	0.560

Table 4.2: Unpaired t-test results for simulated and measured power consumption values for ECOFEN's complete model

Chapter 5

Implementation

Here I describe about SimGrid implementation of energy consumption models.
The challenges encountered
How we tackled the challenges
The limitation of the implementation

Chapter 6

Evaluation

After conducting experiments both on SimGrid and other Simulators, here we compare the result

Chapter 7

Discussion

At this point, you will have some insightful thoughts on your implementation and you may have ideas on what could be done in the future. This chapter is a good place to discuss your thesis as a whole and to show your professor that you have really understood some non-trivial aspects of the methods you used…

Chapter 8

Conclusions

Time to wrap it up! Write down the most important findings from your work. Like the introduction, this chapter is not very long. Two to four pages might be a good limit.

Bibliography

[1] BARAKAT, C., THIRAN, P., IANNACCONE, G., DIOT, C., AND OWEZARSKI, P. Modeling Internet backbone traffic at the flow level. *IEEE Trans. Signal Processing 51*, 8 (2003), 2111–2124.

[2] BARROSO, L. A., CLIDARAS, J., AND HÖLZLE, U. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition.* Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2013.

[3] BARROSO, L. A., AND HÖLZLE, U. The case for energy-proportional computing. *IEEE Computer 40*, 12 (2007), 33–37.

[4] BEISTER, F., DRÄXLER, M., AELKEN, J., AND KARL, H. Power model design for ICT systems - A generic approach. *Computer Communications 50* (2014), 77–85.

[5] BOLLA, R., BRUSCHI, R., DAVOLI, F., AND CUCCHIETTI, F. Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Surveys and Tutorials 13*, 2 (2011), 223–244.

[6] BOLZE, R., CAPPELLO, F., CARON, E., DAYDÉ, M. J., DESPREZ, F., JEANNOT, E., JÉGOU, Y., LANTERI, S., LEDUC, J., MELAB, N., MORNET, G., NAMYST, R., PRIMET, P., QUÉTIER, B., RICHARD, O., TALBI, E., AND TOUCHE, I. Grid'5000: A large scale and highly reconfigurable experimental grid testbed. *IJHPCA 20*, 4 (2006), 481–494.

[7] CARNEIRO, G., FORTUNA, P., AND RICARDO, M. Flowmonitor: A network monitoring framework for the network simulator 3 (NS-3). In *4th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS '09, Pisa, Italy, October 20-22, 2009* (2009), p. 1.

[8] Casanova, H., Giersch, A., Legrand, A., Quinson, M., and Suter, F. Versatile, scalable, and accurate simulation of distributed applications and platforms. *J. Parallel Distrib. Comput. 74*, 10 (2014), 2899–2917.

[9] Claffy, K., Miller, G., and Thompson, K. The nature of the beast: Recent traffic measurements from an Internet backbone. In *Proceedings of INET* (1998), vol. 98, pp. 21–24.

[10] Cornea, B. F., Orgerie, A., and Lefèvre, L. Studying the energy consumption of data transfers in clouds: the ecofen approach. In *3rd IEEE International Conference on Cloud Networking, CloudNet 2014, Luxembourg, Luxembourg, October 8-10, 2014* (2014), pp. 143–148.

[11] Dayarathna, M., Wen, Y., and Fan, R. Data center energy consumption modeling: A survey. *IEEE Communications Surveys and Tutorials 18*, 1 (2016), 732–794.

[12] Fan, X., Weber, W., and Barroso, L. A. Power provisioning for a warehouse-sized computer. In *34th International Symposium on Computer Architecture (ISCA 2007), June 9-13, 2007, San Diego, California, USA* (2007), pp. 13–23.

[13] Fiandrino, C., Kliazovich, D., Bouvry, P., and Zomaya, A. Y. Performance metrics for data center communication systems. In *8th IEEE International Conference on Cloud Computing, CLOUD 2015, New York City, NY, USA, June 27 - July 2, 2015* (2015), pp. 98–105.

[14] Gyarmati, L., and Trinh, T. A. How can architecture help to reduce energy consumption in data center networking? In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy 2010, Passau, Germany, April 13-15, 2010* (2010), pp. 183–186.

[15] Heddeghem, W. V., Lambert, S., Lannoo, B., Colle, D., Pickavet, M., and Demeester, P. Trends in worldwide ICT electricity consumption from 2007 to 2012. *Computer Communications 50* (2014), 64–76.

[16] Jain, R. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.* Wiley professional computing. Wiley, 1991.

[17] Kliazovich, D., Bouvry, P., and Khan, S. U. Greencloud: A packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing 62*, 3 (2012), 1263–1283.

[18] LLC, M. NS-3 simulator, 2017.

[19] Mahadevan, P., Banerjee, S., and Sharma, P. Energy proportionality of an enterprise network. In *Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking 2010, New Delhi, India, August 30, 2010* (2010), pp. 53–60.

[20] Mahadevan, P., Sharma, P., Banerjee, S., and Ranganathan, P. A power benchmarking framework for network devices. In *NETWORKING 2009, 8th International IFIP-TC 6 Networking Conference, Aachen, Germany, May 11-15, 2009. Proceedings* (2009), pp. 795–808.

[21] Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., and Wetherall, D. Reducing network energy consumption via sleeping and rate-adaptation. In *5th USENIX Symposium on Networked Systems Design & Implementation, NSDI 2008, April 16-18, 2008, San Francisco, CA, USA, Proceedings* (2008), pp. 323–336.

[22] Orgerie, A., Lefèvre, L., Lassous, I. G., and López-Pacheco, D. M. ECOFEN: An end-to-end energy cost model and simulator for evaluating power consumption in large-scale networks. In *12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WOWMOM 2011, Lucca, Italy, 20-24 June, 2011* (2011), pp. 1–6.

[23] Shehabi, A., Smith, S., Horner, N., Azevedo, I., Brown, R., Koomey, J., Masanet, E., Sartor, D., Herrlin, M., and Lintner, W. United states data center energy usage report. *Lawrence Berkeley National Laboratory, Berkeley, California. LBNL-1005775 Page 4* (2016).

[24] Sivaraman, V., Reviriego, P., Zhao, Z., Sánchez-Macián, A., Vishwanath, A., Maestro, J. A., and Russell, C. An experimental power profile of energy efficient ethernet switches. *Computer Communications 50* (2014), 110–118.

[25] Sivaraman, V., Vishwanath, A., Zhao, Z., and Russell, C. Profiling per-packet and per-byte energy consumption in the netfpga gigabit router. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (April 2011), pp. 331–336.

[26] Swain, A. R., Hansdah, R. C., and Chouhan, V. K. An energy aware routing protocol with sleep scheduling for wireless sensor networks. In *24th IEEE International Conference on Advanced Information Networking and Applications, AINA 2010, Perth, Australia, 20-13 April 2010* (2010), pp. 933–940.

[27] team, S. SimGrid simulator, 2017.

[28] Velho, P., Schnorr, L. M., Casanova, H., and Legrand, A. On the validity of flow-level tcp network models for grid and cloud simulations. *ACM Trans. Model. Comput. Simul. 23*, 4 (2013), 23:1–23:26.

[29] Weingärtner, E., vom Lehn, H., and Wehrle, K. A performance comparison of recent network simulators. In *Proceedings of IEEE International Conference on Communications, ICC 2009, Dresden, Germany, 14-18 June 2009* (2009), pp. 1–5.

Appendix A

First appendix

This is the first appendix. You could put some test images or verbose data in an appendix, if there is too much data to fit in the actual text nicely.

For now, the Aalto logo variants are shown in Figure A.1.

Aalto University
School of Science

(a) In English

Aalto-yliopisto
Perustieteiden
korkeakoulu

(b) Suomeksi

Aalto-universitetet
Högskolan för
teknikvetenskaper

(c) På svenska

Figure A.1: Aalto logo variants