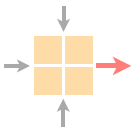
ETH zürich



Networked Systems
ETH Zürich — seit 2015

Power Modelling Framework for Network Switches

Master's Thesis

Author: Jackie Lim

Tutor: Dr. Romain Jacob

Supervisor: Prof. Dr. Laurent Vanbever

July 2023 to February 2024

Acknowledgments

I would like to thank Prof. Laurent Vanbever, Dr. Romain Jacob, and the Networked Systems Group for granting me the opportunity to work on this project. I also want to express my deepest gratitude to my supervisor Dr. Romain Jacob, for his feedback and guidance during the project. Finally, I would like to thank my family and friends for their support.

Abstract

The effective energy consumption of various network devices is currently poorly understood. As a result, it is unclear whether it is possible to optimise the energy usage of network equipment in the context of sustainability. This thesis helps to address this issue by proposing a framework that characterizes the power draw of network switches through experimental methodologies. By applying the framework to 3 commercial data centre switches, trade-offs between performance and energy are identified. Furthermore, by collecting more power data from different network devices, energy-related trends may be discovered.

Contents

Chapter 1

Introduction

1.1 Motivation

Adapting Internet technologies to be more energy efficient is becoming an increasingly important topic. Optimising network equipment to consume less energy helps address this issue. However, there is currently a lack of understanding of the actual power and energy consumption of various network devices.

What poses a bigger problem is how we can acquire such power data. Most often, datasheets only provide the maximum power data required for dimensioning the power supply, even though the actual power draw is typically much lower. Nonetheless, they provide a rough estimation of what we can expect, but for a fine-grained analysis of the power usage, we need a more in-depth process to obtain this information. One way to do this is to design a power analysis framework for network devices, which this thesis is dedicated to.

Concretely, for this project, we set ourselves with the following constraints:

1. With the framework, we want to answer the following questions: Given a specific traffic load, how much power does a given L2 switch draw? Can we optimize it to consume less energy?

2. We want to design a framework that can be applied to a large set of network switches. We envision the framework to be used to collect power data which are then shared in a public database.

3. The framework should be easy to set up. Therefore, we want to avoid requiring specific hardware for the framework (mainly hardware-based traffic generators used in prior work).

1.2 Task and Goals

There are two ideas for designing the framework to meet the constraints. The first idea is to design an energy-related benchmark, and the second is the design of a power modelling framework. Of these two options, we decided on the modelling approach. Both subtly differ in answering the questions in the first constraint mentioned above, which we discuss next.

To be precise, the first approach is to design a benchmark whose main objective is to compare energy-related metrics of network devices in a competitive sense and assess their impact in a real system. That is, we design a benchmark with a set of carefully selected metrics and profiles that we run on our devices under test (DUTs). The results of the benchmark then allow us then to compare the key performance indicators of the DUTs. For instance, *switch A is 20% more energy*

efficient than switch B for IPv6 workload. This type of benchmark helps us answer questions such as *how energy efficient is this device?* or *which device is best for my workload?* On top of that, such benchmarks might incentivize vendors to develop more energy-efficient network equipment.

The second answer is to design a power modelling framework whose primary focus is providing a comprehensive power characterisation of a network device. One way to do this is to propose a power model and determine its parameters through specific methodologies. It allows us to get a better understanding of what component contributes how much to the total power of the network device, and could also encourage network operators to configure their deployed network equipment in a more energy-efficient manner. In other words, this type of benchmark would answer *what if*-questions such as *how much energy could we save, if we were to configure switch A this way?*

Of these two approaches, we are focusing on the power modelling framework since it is a missing piece for network-wide energy optimization. Hence, the main contributions of this work are the following:

- We present a power model intended for L2 switches and define methodologies to determine the model parameters robustly.
- We implement the framework using an external power measurement tool, as well as one server running Cisco TRex Traffic Generator fuelled by Data Plane Development Kit (DPDK).
- We test the framework process by applying it to three commercial switches intended for data centres and derive their respective model parameters.

1.3 Overview

The report is organised as follows: Chapter 2 provides insights into related work. In Chapter 3 we present the power model and the methodology to derive the model parameters. In Chapter 4 we then describe the implementation of our framework. The evaluation of the DUTs is presented in Chapter 5. Finally, we discuss the topic of future work in Chapter 6.

Chapter 2

Related Work

This chapter provides background on power measurement and power modelling of network devices in Section 2.1. We then elaborate in Section 2.2 on existing benchmarks dedicated to network devices. This includes the choices of energy-related metrics and methodologies.

2.1 Power Measurement and Power Modelling

Several power measurements of switches and routers have been done by academic groups. Some of these groups have also proposed an empirical power model based on their findings.

For example, the authors of [29] have studied the energy consumption of a NetFPGA-based router. And in our previous work, we profiled the power usage of an Intel Tofino switch [31]. The measurement results show similar findings that the energy consumption of network devices varies depending on the number of active ports, the amount of traffic load, and the packet sizes within the load. Moreover, specifically for the Intel Tofino switch, the power varies with the complexity of the data plane program.

Furthermore, authors from [24] and [37, 38, 36] have measured the power draw of network devices and proposed power models based on the traffic load. These models share the common properties that on top of the idle power, which we refer to as static power in this report, two factors affect the power draw of a network device. Specifically, in a traffic load scenario, there is an energy cost for processing a bit, and additionally, there is a cost for processing a packet.

In this thesis, the power model consists of two major parts, namely the static and the dynamic power. The dynamic power, in which traffic load is processed, is based on the models from [24, 37, 38, 36], whereas the static power is based on the measurement results from [29, 31]. The details of the power model are further explained in Chapter 3.

2.2 Network Device Benchmark

Currently, there exist several proposals on benchmark methodologies for network devices in the form of RFCs [18, 3, 26, 30, 28]. Note that these RFCs provide information to the Internet community, and do not specify an Internet standard of any kind. Moreover, these RFCs assess the performance of the main functionalities of network devices and quantify them; thus, some of the technical reports share similar metrics such as throughput, latency, or frame loss rates. On top of that, further definitions of metrics are extended in other technical reports [25]. However, these reports did not consider any energy metrics, as the topic of sustainability was not their primary focus.

As a remark, there exists an expired RFC draft from 2013 about benchmarking power usage of network devices [34]. And currently, another RFC draft is written regarding the topic of Power and Energy Efficiency [32].

Mahadevan et al. [33] published a power benchmark framework that assesses the efficiency of network devices by evaluating the power draw based on the device's utilisation. However, it seems that the framework has not been published. Furthermore, standardization committees such as the ETSI [12], the ITU-T [10, 13], and the ATIS [17] have specified methodology procedures and metrics to assess the energy efficiency of network devices which we discuss next.

Metric

The authors of [27] have presented an excellent summary of various energy-related metrics including those of Mahadevan et al., [33] the ITU-T [10, 13], and the ATIS [17].

In particular, most of the metrics describing the energy efficiency of network devices quantify the efficiency as a value which relates the units of watt and Gbps together. One example is the metric called Energy Consumption Rate, or ECR for short. According to ITU-T this metric describes the power required to move one gigabit worth of line-level data per second and is therefore expressed in units of $\frac{W}{Gbps}$. There are other variations of this metric such as the Energy Efficiency Rate, or EER, which is simply the inverse of the ECR. Moreover, the ATIS specification, this metric is also called TEER, or Telecommunication Energy Efficiency Rating, which describes the ratio of "useful work" to power consumption.

The problem with ECR is that it is determined from a single power measurement setup at maximum load. It is derived by dividing the measured power by the maximum throughput in that measurement. Hence, any information about the static power when no load is applied is not considered. Concretely, we do not know how much power a device uses in its idle state where no traffic is involved.

To address this issue the ITU-T specifies another variation of the ECR which is called the Weighted Energy Consumption Rate, or ECRW. This metric uses additional weighting terms that reflect the expected workload (including the idle state) of the device class to represent a more realistic Energy Consumption Rate value.

Furthermore, Mahadevan et al. define an energy metric called the Energy Proportionality Index or EPI, which describes the energy proportionality based on the energy consumption at idle and peak utilization. It is expressed in percentages. A device with an EPI of 100% essentially means that all of its power usage is load-dependent. Meanwhile, a device with an EPI of 0 means that its total power usage is load-independent and is constant. However, as mentioned by Mahadevan et al., the EPI does not necessarily translate into energy efficiency, as this metric alone does not contain information about how much power is effectively required for a given load. Thus, Mahadevan et al. additionally use the ECR metric (called NormalizedPower) in their benchmark.

Another problem with the ECR metric is the fact that from previous works mentioned in Section 2.1 there is an observation that the power of network devices scales over both bit rate and packet rate. ECR only considers the bit rate and ignores any information about the packet rate.

Finally, we note that the metrics described here are more appropriate for an energy benchmark, rather than for a power modelling framework. For our project, instead of evaluating metrics, we determine power model parameters for each of our DUTs.

Methodology

Another aspect to consider is how to send high-volume traffic such that the switch is sufficiently utilized. This can be challenging given that modern switches can handle multiple ports at 100 Gbps line rate. Most of the related work mentioned previously in Section 2.1 relied on hardware-based traffic generation tools to generate load over a set of ports. In this thesis we avoid this, however, we do require Network Interface Cards (NIC) that utilize DPDK [8] to generate high-volume traffic.

Other than that, Mahadevan et al. used a less expensive method through a loopback technique by assigning a set of loopback ports on the DUT: Two ports are reserved to receive and return loads from traffic generators. By generating traffic with broadcast addresses, the generated traffic is broadcast within the DUT which then ensures that all loopback ports are utilized. In our work, we use a similar method that requires loopback ports, although we do not rely on generating traffic with broadcast addresses to distribute the load among the ports. To be more precise, in our project, we send the load along a snake-like path over the DUT, which requires loopback ports and VLAN functionality on the network switch. In fact, the method for performing this so-called snake test comes from the RFC 8239 [26], which we describe further in Section 3.2.

Chapter 3

Design

This chapter presents the power modelling framework. This includes the power model in Section 3.1, and the methodology for determining the power model parameters in Section 3.2. A list of recommended points to be reported for the measurement is described in Section 3.3. Finally, we discuss the granularity of the power model in Section 3.4. This includes a discussion of parameters we thought of including in the model and why we decided not to.

3.1 The Power Model

The power model of an L2 switch is based on similar findings from previous power measurements described in Section 2.1 and additional empirical measurement results during this project. Our assumption is that the switch is a single-linecard device. We describe the total power of the switch using the following equations:

In short, Eq. (3.1) describes the total power of the switch that is based on the static and dynamic power. The former, $P_{sta}(C)$, depends on the device configuration, denoted by C and the latter, $P_{dyn}(C, L)$, depends on both device configuration and the total traffic load denoted by L.

$$P_{switch} = P_{sta}(C) + P_{dyn}(C, L) \tag{3.1}$$

$$P_{sta}(C) = P_{base} + \sum_{i=0}^{N_{port,active}} P_{port,sta}(c(i)) \tag{3.2}$$

$$P_{port,sta}(c(i)) = P_{port,switch}(c(i)) + P_{trx}(c(i)) \tag{3.3}$$

$$P_{trx}(c(i)) = P_{trx,plugged} + P_{trx,active}(c(i)) \tag{3.4}$$

$$P_{dyn}(C, L) = \sum_{i=0}^{N_{port,active}} (P_{port,dyn}(c(i), l(i)) + P_{port,offset}(c(i))) \tag{3.5}$$

$$P_{port,dyn}(c(i), l(i)) = E_b(c(i)) \cdot r(i) + E_p(c(i)) \cdot p(i) \tag{3.6}$$

Furthermore, Eqs. (3.2), (3.3) and (3.4) describe the static power and Eqs. (3.5) and (3.6) describe the dynamic power respectively in more detail which we explain in the following.

Static power: Eqs. (3.2), (3.3) and (3.4)

The static power in Eq. (3.2) consists of two parts, namely, the base power of the device P_{base}, and the power of the active ports described as a sum $\sum_{i=0}^{N_{port,active}} P_{port,sta}(c(i))$. In this report, we refer to an active port as an interface that is enabled and in a state where it is ready to send and receive traffic. Depending on the configuration of an active port i, e.g., its line rate, denoted as $c(i)$, the power cost of that active port may vary.

To be more precise, the cost of an active port is modelled by two components in Eq. (3.3): There is a portion of the power cost that comes from the switch side $P_{port,switch}(c(i))$, and a second portion that is required to power the transceiver $P_{port,trx}(c(i))$, which may be zero in the case of passive transceivers.

The transceiver power is described in Eq. (3.4) where it consists of two components, mainly, in the case where we simply plug it into the network device $P_{trx,plugged}$, and a second part $P_{trx,active}(c(i))$ where it is enabled while its corresponding port is active. This equation is based on results we observed in our previous work [31], as well as on empirical results we observed during this project.

The details of why we choose to model the transceiver part are discussed in Section 3.4.

Dynamic power: Eqs. (3.5) and (3.6)

The dynamic power is shown in Eq. (3.5) and is based on the work in Section 2.1. It describes the power as a function on the active ports that are currently forwarding traffic load denoted as $l(i)$. To be more precise, in Eq. (3.6) we have for each active port i an energy cost to process one bit $E_b(c(i))$ multiplied by the sum of the input and output bit rates $r(i)$ on that particular port. Similarly, we model an additional energy cost to process a packet $E_p(c(i))$ multiplied by the summed input and output packet rate $p(i)$ at port i.

$E_b(c(i))$ represents the energy cost of sending or receiving bits, while $E_p(c)$ is the energy required to process a frame over the interface. This includes header processing tasks, as well as table lookups. Note that these energy costs can vary depending on the configuration of the port $c(i)$.

The last term $P_{port,offset}(c(i))$ in the dynamic power describes a power offset, when low traffic is being processed at port i. This term is added to the model based on empirical results we observed in our measurements. This is discussed further in the evaluation in Chapter 5.

Lastly, we note that there is no transceiver power in $P_{dyn}(C, L)$ as we assume the transceiver power is constant and independent of load. This is further explained in Section 3.4.

3.2 Methodology

To determine the power model parameters, we require a tool that measures the total power of the DUT. On top of that we also require traffic generation to evaluate the dynamic power parameters. More detail about the tools we used is described in Chapter 4. If plug-in transceivers are used, we assume all transceivers to be of the same type. This is not a fundamental limitation but it simplifies the measurement methodology.

To determine all the previously mentioned power model parameters. We propose the following measurement methods:

Static power: Eq. (3.2)

Unless specified otherwise, we assume that the traffic ports are already physically connected to other endpoints on another switch.

As a reminder, we consider a port to be active if it's enabled and in a state where it is ready to send and receive traffic. Otherwise, a port is considered to be inactive. In this case, the port must be shut down, and if this port is connected to another endpoint, then that endpoint must also be shut down.

P_{base}: We measure the power of the device in a state where all ports are inactive. If traffic ports feature transceiver modules, disconnect them (as they are considered in term $P_{trx,plugged}$ described later). Management interfaces are allowed to be active. The base power P_{base} corresponds to the measured value.

$P_{port,sta}(c_j)$: This power represents the cost of an active port configured with c_j. We measure the power of the device over several iterations, where in each iteration we configure a certain number of active ports equally with c_j. This means that for a switch with n connected ports configured with c_j, we measure the device's power with $0, 1, 2, ..., n-1, n$ active ports.

Another method is to connect ports in pairs in loopback and measure the power with $0, 2, 4, ..., n-2, n$ active ports configured with c_j, where each pair of loopback ports is either active or inactive. This allows us to determine $P_{port,sta}(c_j)$ without the need for a second connected switch.

We relate the total power of the device to the number of active ports and then use linear regression to determine the cost of an active port configured with c_j. The slope of the linear fit represents the cost $P_{port,sta}(c_j)$.

Regarding the concrete measurement procedure, we randomise the order in which we activate the ports (or pairs respectively) over the iterations. For instance, in the first iteration, we activate 4 ports, in the second we activate 0 ports, in the third we activate n ports, etc. Note that we also randomise which ports (or pairs) we activate within the iteration.

Static power including Transceivers: Eqs. (3.3) and (3.4)

In the case of transceivers, we require additional steps to determine its power portion $P_{trx}(c(i))$ which is discussed next:

$P_{trx,plugged}$: This term represents the power cost of one transceiver if plugged into the switch and inactive. The setup is similar to measuring P_{base} where all ports must be inactive. The only difference is that we additionally plug a set of n equal transceiver types to the DUT and connect them to other endpoints. Note that the transceiver interfaces must also be inactive. We measure the power of the device and subtract the measured value from the previously measured base power P_{base}. This in turn gives us the power required for powering n transceivers when plugged. Finally, we divide this power by n to determine $P_{trx,plugged}$.

$P_{trx,active}(c_j)$: This term represents the transceiver power in an active port configured with c_j. To determine this value, a second switch with compatible transceiver interfaces is required. To determine this parameter we first perform the measurement to determine $P_{port,sta}(c_j)$ as described above (this part can be done without a second device by looping ports in pairs).

We then repeat the same measurement procedure to determine $P_{port,sta}(c_j)$ on our DUT with the second device. The only difference is that during the iterations, all connected ports of the second device are always disabled, regardless of whether we enabled or disabled a port on the DUT.

This way we determine the cost of a port from the switch side $P_{port,switch}(c_j)$. This is because the transceiver cannot communicate with its other endpoint (as we shut them down) and is therefore

not active.

From the total cost of a port $P_{port,sta}(c_j)$ we subtract the port cost from the switch side $P_{port,switch}(c_j)$, which in turn gives us the power of the transceiver $P_{trx,active}(c_j)$. Hence, $P_{trx,active}(c_j) = P_{port,sta}(c_j) - P_{port,switch}(c_j)$. [1]

Dynamic power: Eqs. (3.5) and (3.6)

$P_{port,dyn}(c_j, l(i))$: In order to derive the energy per bit $E_b(c_j)$ and energy per packet $E_p(c_j)$ parameter for a given port i with configuration c_j, we use a similar derivation presented by Vishvanath et al. [38]. For the remainder of this report, we use the terms packet and frame interchangeably, as the focus of this project is about Layer 2 switches. Note that in this report we use the word packet to refer to an L2 frame.

We run multiple iterations of measurements, utilising a set of equally configured ports, by forwarding traffic load over a range of traffic rates, and over multiple frame sizes. We first present the derivation of the energy parameters $E_b(c_j)$ and $E_p(c_j)$ for port i configured with c_j:

$$P_{port,dyn}(c_j, l(i)) = E_b(c_j) \cdot r(i) + E_p(c_j) \cdot p(i) \tag{3.7}$$

$r(i)$ corresponds to the summed input and output L1 bit rate whereas $p(i)$ corresponds to the summed input and output L2 frame rate at port i. Thus, $p(i)$ and $r(i)$ are related by the equation:

$$p(i) = \frac{r(i)}{8 \cdot (L + 20)} \tag{3.8}$$

L is the frame size in bytes, and the 20 additional bytes come from the preamble, interpacket-gap, and start frame delimiter [21]. This is required since we relate L1 rate to L2 frame sizes. Plugging Eq. (3.8) into Eq. (3.7) we obtain:

$$P_{port,dyn}(c_j, l(i)) = E_b(c_j) \cdot r(i) + E_p(c_j) \cdot \frac{r(i)}{8 \cdot (L + 20)} \tag{3.9}$$

The first derivative with respect to $r(i)$ returns

$$\frac{\partial P_{port,dyn}(c_j, l(i))}{\partial r(i)} = E_b(c_j) + \frac{E_p(c_j))}{8 \cdot (L + 20)} \tag{3.10}$$

By keeping the frame size L constant, we assume that by increasing the rate $r(i)$, the dynamic power of the port scales linearly. In other words, we assume that for a fixed L, the dynamic power has the form:

$$P_{port,dyn}(c_j, l(i))_L = a_L \cdot r(i) + b_L \tag{3.11}$$

and therefore,

$$\frac{\partial P_{port,dyn}(c_j, l(i))_L}{\partial r(i)} = a_L \tag{3.12}$$

Note that the coefficient a_L is derived experimentally by measuring the power of the device for a range of rates and for a fixed frame size L, and is then estimated with a linear fit. Eqs. (3.10) and (3.12) are the same, thus:

$$E_b(c_j) + \frac{E_p(c_j)}{8 \cdot (L + 20)} = a_L \tag{3.13}$$

[1] To determine $P_{trx,active}(c_j)$, we use formula $P_{trx,active}(c_j) = P_{port,sta}(c_j) - P_{port,switch}(c_j)$ described in the methodology. We are aware that this equation does not correspond exactly to Eqs. (3.3) and (3.4) in the power model (due to the missing $P_{trx,plugged}$). However, we describe the model in Section 3.1 this way because we prefer to group the transceiver power portion together in one term.

or equivalently:

$$8 \cdot L \cdot E_b(c_j) + 8 \cdot 20 \cdot E_b(c_j) + E_p(c_j) = 8 \cdot (L + 20) \cdot a_L \tag{3.14}$$

This equation describes the total energy consumed per frame as a function of the frame size L. The left hand side is a linear function in L where $8 \cdot E_b(c_j)$ is the slope, and $8 \cdot 20 \cdot E_b(c_j) + E_p(c_j)$ is the intercept. The right hand side is derived experimentally by selecting a set of frame sizes L, and determining the coefficient a_L for each frame size described in Eq. (3.12). Hence, on the right-hand side, we plug in L and the corresponding estimated a_L and obtain several data points. To determine the slope and the intercept on the left hand side in Eq. (3.14) we use linear regression. And from this, we have two equations to derive the two unknown energy parameters $E_b(c_j)$ and $E_p(c_j)$. In Appendix A we show some plots visualizing the derivation of the energy parameters from one of our DUTs as an example.

Note that this derivation is the case for one port i configured with c_j. Since we describe the dynamic power as a sum of ports, this is not a problem, as we can send traffic over multiple equally configured ports with c_j and divide the measured slope a_L by the number of utilised ports to get the coefficient for a single port.

Moreover, the traffic should indeed be sent over multiple, equally configured ports of the DUT to ensure that we can measure a larger power difference over the range of traffic rates to get a better estimate of a_L. To ensure this, we use the so-called snake test described in RFC 8239 [26]. This setup allows us to generate bi-directional traffic over two ports, which is then forwarded over a set of ports configured with c_j on the DUT. In this report, we refer to this as the snake test. From the RFC 8239:

> Alternatively, when a traffic generator cannot be connected to all ports on the DUT, a snake test MUST be used for line-rate testing, excluding latency and jitter, as those would become irrelevant. The snake test is performed as follows:
>
> - Connect the first and last port of the DUT to a traffic generator.
>
> - Connect, back to back and sequentially, all the ports in between: port 2 to port 3, port 4 to port 5, etc., to port N-2 to port N-1, where N is the total number of ports of the DUT.
>
> - Configure port 1 and port 2 in the same VLAN X, port 3 and port 4 in the same VLAN Y, etc., and port N-1 and port N in the same VLAN Z.
>
> This snake test provides the capability to test line rate for Layer 2 and Layer 3 [RFC2544] [RFC3918] in instances where a traffic generator with only two ports is available. Latency and jitter are not to be considered for this test.

As for the concrete measurement, we measure the power of the device in iterations where we send a given load over a range of the Cartesian product of frame sizes and bit rates in the snake test. In our project, we used a range of 10 rates, and 10 frame sizes depending on the line rate configuration of the ports. All utilised ports are configured equally with c_j. And with the previously described derivation, we determine the energy parameters $E_b(c_j)$ and $E_p(c_j)$.

$P_{port,offset}(c_j)$: This term describes an offset of the dynamic power for a port configured with c_j when low traffic is processed on that particular interface. We observed for our DUTs that when deriving the slope a_L, the data point at 0 Gbps did not always match the linear behaviour in Eq. (3.12). We replace the power at 0 Gbps with the power at which we send small amounts of traffic and measure the difference as $P_{port,offset}(c_j)$. [2] The power at this rate fits better with the linear behaviour of the rate and adds another potential data point to derive the coefficient a_L. More details on this observation is shown in the evaluation in Chapter 5.

Concretely, the offset is derived during the snake test where we additionally add 2 more data points to the measurement. That is, including the 100 data points from the Cartesian product of traffic rates and frame sizes, we also measure the power of the device where no traffic is sent, and where low traffic, e.g., 1 frame per second (pps) for a frame size of 1500 bytes, is sent. The difference in power between these two cases divided by the number of ports used in the snake test corresponds to the port offset $P_{port,offset}(c_j)$. And as previously mentioned, we replace the power at 0 Gbps with the power at 1 pps and use it as an additional data point to determine all a_L.

3.3 Report

It is important to be able to replicate the evaluated parameters. Hence, we recommend reporting the following points:

- The type of transceiver, if used, and the type of cables used to connect the DUT.
- Other configurable hardware characteristics of the DUT, e.g., fan or power modules.
- The ambient temperature during the measurement.

More details and why we recommend these points are discussed in the next section.

3.4 Granularity of the Model

The difficulty in choosing a power model is to find a model that is accurate enough to give us information about the effective power draw of the switch, and at the same time, is applicable to a large set of DUTs. Having a too fine-grained power model may also increase the burden on the user of the framework as it might require more measurements that could be harder to set up, given that we measure the total power of the device with an external power meter. One example is discussed next.

MAC table entries

One idea we initially had in mind for our model was to add a term describing the static power attributed to the MAC table entries. At the start of the project, we assumed that it would scale linearly with the number of entries. However, the reason why we decided not to include it in the benchmark is that later on, we believe this power portion to be constant and independent of the number of entries, as all table entries in memory are always powered on. And even if there were a power change depending on the number of entries, the difference would be too small to be effectively measured with an external power meter tool. On top of that, it would require additional setup procedures where we had to fill the MAC table entries. This depends on the size of the

[2] We believe that this assumption is also valid in real systems, as network switches would normally always see at least some background control traffic.

table, which varies among devices and may not even be known by the user. Therefore, it could potentially require additional steps to first find out how large the MAC table is which could be too cumbersome.

Transceiver power portion

In our static power model analysis, we include the transceiver power as a separate term $P_{trx}(c(i))$, because the focus of our power model is the switch. If the switch features transceivers, then the measured power may differ depending on the type of the transceiver modules. Generally speaking, optical transceivers require much more power than electrical transceivers. Not separating the transceiver power from the interface power may give us a false impression of the actual cost of a port for a given switch.

For example, according to the specification of FLEXOPTIX, the total power draw of their passive 100G Direct Attach Cable (DAC) QSFP28 transceiver is 0.06 W [2]. On the other hand, the power draw for a 100G Active Optical Cable (AOC) QSFP28 transceiver is 7 W [1].

The main difficulty in separating the transceiver power from the total interface power is that there is no proper method to measure the transceiver power in isolation, as the measured entity, the switch, includes both port and transceiver. Therefore, we choose to add additional equations and methodologies to estimate the transceiver power. However, we note that the methodologies to determine the transceiver power may not apply correctly to all switches since it depends on the circuitry of the device.

Unlike the static power, it becomes much more difficult when traffic load is involved. In the case of dynamic power, we do not know of any method of obtaining an estimate of the transceiver power by measuring it externally. Thus, our assumption of the transceiver power is that it is constant and independent of the load, and we only explicitly include the transceiver power in the static power.

Fan and Power Supply Units

Other subjects we have considered to factor in our model are the fan and the power supply units (PSU), as both components have a non-negligible impact on the measured switch power. Furthermore, modern data centre switches feature hot-swappable fan modules and power supply modules. Depending on which particular module the switch features, its power contribution can change.

As an example for the fan power of the Wedge100BF-32X [7] network switch, we received data from the vendor, that the power for a single fan can reach up to 18.84 W at maximum revolutions per minute (rpm). The device features 5 of these fan modules, whose rpm ultimately depend on the utilisation of the switch.

As for the power supply unit, or PSU, we are interested in its efficiency. Depending on the load of the powered device, the PSU may draw more power than the device actually needs. Certification program like 80 Plus exist to promote energy efficiency in power supply units [20]. Including the efficiency in our model could help us understand how much the switch effectively requires. Especially in the case where data centre switches have multiple power supply unit modules for redundancy. In such a scenario, the loads on both power supply units are lower than in the case where we only use one PSU module, and lower load typically means lower efficiency.

We decided not to include the fan power and the PSU efficiency in our model because we are not interested in this level of granularity, as the main focus of our power model lies in the port configuration. In other words, our power model aims to help optimise energy consumption based on the interface configuration.

Assuming we included the fan power in the model and acquired data for some switches, it would be interesting to investigate whether we can optimise its power draw based on its fan power, i.e., by cooling the ambient temperature, especially if it is unclear how much additional power it would require to cool the room. However, this topic lies beyond the scope of this project. Contrary to that, we are interested in understanding how we can optimise energy efficiency through configuring the interfaces appropriately.

On top of that, getting the data for the PSU efficiency and modelling the fan power of a switch can be challenging. Thus, we recommend reporting the modules, especially if they are swappable, as well as the device or ambient temperature during the measurement.

Chapter 4

Implementation

This chapter discusses the lab setup for conducting the power modelling experiments in Section 4.1, and the measurement overview including our DUTs in Section 4.2.

4.1 Lab Setup

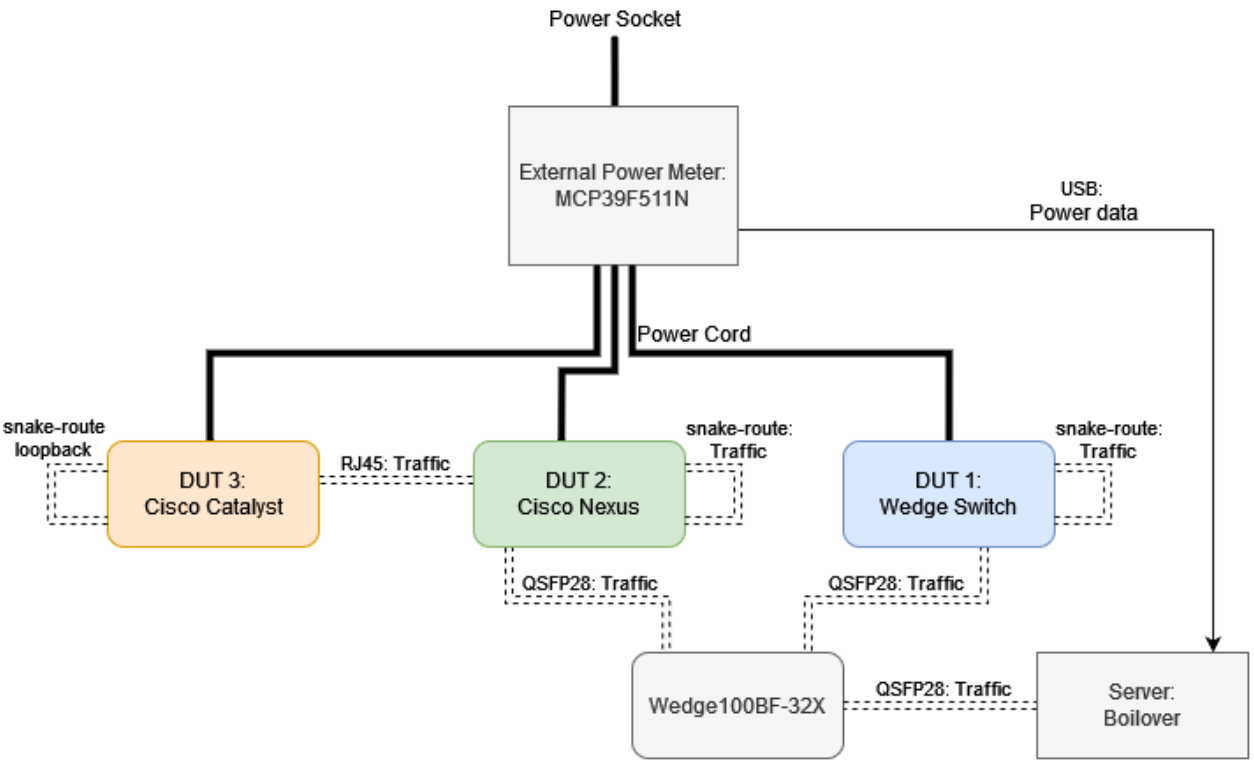


Figure 4.1: Final lab setup. Note that at the start of the project, of all DUTs, we only had the Wedge Switch DUT in the lab setup, which we measured first. Later, we added the Cisco Nexus DUT to the lab setup on which we applied the power modelling separately. Finally, we added the Cisco Catalyst DUT to the setup for its power measurement.

Our lab setup shown in Fig. 4.1 includes an MCP39F511N power monitor board [11], an external power meter which we connect to the power socket, to our DUTs, and our server boilover. We

use boilover to read power measurements and to generate high-volume traffic. We also use the Wedge100BF-32X, a P4-programmable switch [7, 9, 15], to forward traffic appropriately to some of our DUTs. In particular, one DUT (a Cisco Catalyst) that does not have QSFP28 interfaces is instead connected through another DUT (a Cisco Nexus) to receive traffic from the server for the snake test.

Finally, the testbed is located in a server room that is separated into two compartments: A cold aisle that is regulated at roughly 19.2 °C for the air intake, and a hot aisle for the exhaust with a measured average temperature of 21.9 °C.

4.1.1 Devices Under Test

During the project, we applied the power modelling on 3 different network switches. These are the Wedge100BF-32X [7], the Cisco Nexus 9000 C93108TC-FX [5], and the Cisco Catalyst WS-C3560-24PS [4]. In this report, we refer to them simply as the Wedge Switch, Cisco Nexus, and Cisco Catalyst. A summary of the relevant properties is shown in Table 4.1.

It should be noted that the Wedge Switch and Cisco Nexus each have two PSU modules, but for the power measurement, we powered these DUTs from one PSU each. Moreover, the Cisco Catalyst features only one PSU.

DUT	Series / Hardware Property	Release Year	Ports
Wedge Switch	Programmable Tofino ASIC [9] 2x PFE600-12-054NA PSU	2017	48x QSFP28
Cisco Nexus	Cisco Nexus 9000 4x NXA-FAN-30CFM-F Fan Module 2x NXA-PAC-500W-PE PSU	2017	48x RJ45 6x QSFP28
Cisco Catalyst	Cisco Catalyst 3560	2004	26x RJ45 2x SFP

Table 4.1: Our DUTs. We report PSU and fan modules if other variants exist for the DUT.

Regarding the QSFP28 transceivers, we used passive Direct Attach Cables (DAC) to connect the QSFP28 interfaces.

Finally, we configured both Cisco devices with multiple VLANs according to the methodology to run the snake test in Section 3.2. On the Wedge Switch, however, we programmed the data plane using P4 [9] and configured the device to statically forward traffic based on which ingress port the traffic came from. Hence, traffic is forwarded along the snake route without requiring any VLAN functionality. The P4 code and the configuration of the Cisco devices are public and can be found in [22].

4.1.2 boilover and Traffic Generation

For the dynamic power analysis, we run Cisco TRex v3.04 on our server boilover, which is equipped with a Mellanox MCX516A-CCAT ConnectX®-5 Network Interface Card (NIC) [14]. The NIC has two interfaces and allows us to generate high-volume bi-directional traffic. We decided on using Cisco TRex since we want to avoid using specialised hardware traffic generation tools for the

framework. For more details on high-volume traffic generation and specifically to Cisco TRex, we refer to [35] and [6] respectively.

For the snake test, we write Python scripts to generate stateless traffic over TRex. The load includes a UDP datagram with fixed, arbitrarily selected source and destination ports (12 resp. 1025) within an IPv4 packet. The source and destination address of the IPv4 packet is set to 2.2.2.2 and 4.4.4.4 respectively, depending on the direction of the traffic flow. Note, however, that for L2 switches this does not matter.

At the Ethernet layer, we set one fixed source address, 2:2:2:2:2:2, and one fixed destination address, 4:4:4:4:4:4, and vice versa depending on the direction of the traffic flow. The frame is additionally padded to generate traffic with specific frame sizes.

Finally, we connect our server boilover to the MCP39F511N Power Monitor Board to read the power data through the software tool PinPoint. The source code for PinPoint can be found here [23].

4.2 Measurement Procedure

Using Python and Bash Script, we coordinate the testbed. Moreover, we manually configure our DUTs remotely via ssh. For the measurement procedure, we followed the methodology described in Section 3.2.

On PinPoint, we set our sampling interval to 50 ms for all measurements. On top of that, the measure time of a measurement iteration is 15 seconds. However, there is an exception with the Cisco Nexus device, where the measurement time is 76 seconds instead. This is due to an observation of a periodic power pattern on the Nexus in which a specific power pattern repeats every 76 seconds. This is further elaborated on in Section 5.1.1. We use the median value of the power data to determine the power for each measurement iteration. Finally, we repeat each measurement 3 times and use the median value to evaluate the results in the next chapter.

For the static power analysis, we select a set of interfaces to determine the power cost of the ports over several iterations following the methodology described in Section 3.2. The same applies to the dynamic power analysis: We select a set of ports utilized during the snake test. Table 4.2 shows a summary of the measurement time, as well as the number of ports we used on our DUTs to run the tests.

DUT	Measurement Time [s]	Port Type	Static Power: # Ports used	Dynamic Power: # Ports used
Wedge Switch	15	QSFP28	12	14
Cisco Nexus	76	QSFP28	6	6
		RJ45	16	32
Cisco Catalyst	15	RJ45	14	16

Table 4.2: Measurement properties summary

Furthermore, we run the snake test with 10 different frame sizes and 10 different traffic rates. Table 4.3 shows the data points we select depending on the line rate on the port configuration. Due to performance limitations on boilover, we cannot send more than 40 Gbps of bi-directional

traffic (that is, 40 Gbps in both directions, for a total rate of 80 Gbps) with frame sizes less than 500 bytes over the NIC.

Line rate	Bi-directional traffic rates	Frame sizes [bytes]
100 Gbps 40 Gbps	{4, 8, 12, 16, 24, 28, 32, 36, 40} Gbps	{500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1500}
25 Gbps	{2.5, 5, 7.5, 10, 12.5, 15, 17.5, 20, 22.5, 25} Gbps	{256, 300, 384, 512, 600, 768, 1024, 1200, 1300, 1500}
10 Gbps 1 Gbps 100 Mbps	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10} Gbps {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1} Gbps {10, 20, 30, 40, 50, 60, 70, 80, 90, 100} Mbps	{64, 96, 128, 192, 256, 384, 512, 768, 1024, 1500}

Table 4.3: Traffic rate and frame sizes for the snake test

Note that the Cisco Nexus receives traffic over QSFP28 ports. For the snake test to evaluate its RJ45 ports, the power of the QSFP28 endpoints must be subtracted from the total measured power. Thus, we first determine the model parameters of the QSFP28 ports, before including them in calculating the dynamic power portion of the RJ45 interfaces. Using our model in Section 3.1 we know the dynamic power portion of the two QSFP28 endpoints.

For the low traffic scenario to determine the offset power $P_{port,offset}(c(i))$, we send a load at 1 pps and 1500 bytes frame sizes except for the Cisco Catalyst DUT, where we instead send a load of 1 Mbps and 1500 bytes. This is because at 1 pps we did not observe any changes in power. We elaborate on this in Section 5.1.1. Finally, within each iteration of the snake test, after we start sending a specific traffic load, we wait for 10 seconds before starting to measure the power.

Chapter 5

Evaluation

Using the power model, performance and energy trade-offs can be identified among different port configurations, as well as among different devices. Moreover, from the power model measurement results of our three DUTs, we observe that:

- The newer DUTs from 2017 have a significantly higher base power than the Cisco Catalyst from 2004. However, in terms of dynamic power, the newer devices are much more energy efficient at processing load than the Catalyst.
- RJ45 ports require more power to maintain than QSFP28 ports for the same line rate.
- For a given interface type and device, lowering the line rate configuration of a port reduces overall energy consumption.
- Moreover we observe a trend for QSFP28 port that for lower line rate configuration, the corresponding interface requires more power to process the bits of a frame, than the frame itself.
- On all devices, we observe a power offset when low traffic is detected. Furthermore, this offset is negative for RJ45 ports and positive for QSFP28 ports.

The power model parameters are shown in Section 5.1. Lastly, we discuss the topic of validation of the model in Section 5.2.

5.1 Power Model Parameters

Wedge Switch

- P_{base}: 108.1 W
- $P_{trx,plugged}$: 0 W

For the static power parameters $P_{port,sta}(c)$, $P_{port,switch}(c)$, and $P_{trx,active}(c)$, we used the data from our previous work [31]. Moreover, for all QSFP28 configurations, we disabled forward error correction (FEC).

c	$P_{port,sta}(c)$	$P_{port,switch}(c)$	$P_{trx,active}(c)$	$P_{port,offset}(c)$	$E_b(c)$	$E_p(c)$
QSFP28: 100G	1.57 W	0.88 W	0.69 W	0 W	1.72 pJ	7.21 nJ
QSFP28: 25G	0.52 W	0.21 W	0.31 W	0.05 W	2.54 pJ	5.62 nJ
QSFP28: 10G	0.31 W	0.21 W	0.1 W	0.06 W	2.66 pJ	4.67 nJ

Table 5.1: Wedge Switch model parameters

Cisco Nexus

- P_{base}: 147.01 W
- $P_{trx,plugged}$: 0.11 W

c	$P_{port,sta}(c)$	$P_{port,switch}(c)$	$P_{trx,active}(c)$	$P_{port,offset}(c)$	$E_b(c)$	$E_p(c)$
QSFP28: 100G	0.4 W	0.17 W	0.23 W	0 W	5.37 pJ	21.19 nJ
QSFP28: 40G	0.23 W	0.07 W	0.16 W	0.03 W	6.54 pJ	17.64 nJ
RJ45: 10G	2.06 W	2.06 W	n/a	-0.03 W	6.86 pJ	16.85 nJ
RJ45: 1G	0.93 W	0.93 W	n/a	-0.03 W	33.83 pJ	18.20 nJ

Table 5.2: Cisco Nexus model parameters

On the configuration QSFP28: 100G we disabled FEC. On the 40G however, we set it to auto since we could not explicitly disable it.

Cisco Catalyst

- P_{base}: 40 W

c	$P_{port,sta}(c)$	$P_{port,switch}(c)$	$P_{trx,active}(c)$	$P_{port,offset}(c)$	$E_b(c)$	$E_p(c)$
RJ45: 100M	0.21 W	0.21 W	n/a	-0.01 W	15.73 pJ	193.13 nJ

Table 5.3: Cisco Catalyst model parameters

5.1.1 Measurement Observation

Grouping the devices in terms of release date, we observe that the base power has changed over time. The oldest device is the Cisco Catalyst from 2004 with a base power of 40W, and the Cisco Nexus and Wedge Switch released in 2017 have a base power of 147.01W and 108.1W respectively. This makes sense as technology has always been pushing in recent years with newer devices featuring smaller electronics and designed with more complexity and high performance.

On the other hand, if we consider the energy costs E_b and E_p, we observe that the older device has a noticeably lower energy efficiency. Especially the energy cost for processing packets which seems to have become more efficient with newer network switches (though it would be great if we could collect more data from various switches to confirm this observation).

Another observation is that if we group by interface type, keeping the RJ45 ports active seems to be more costly than an active QSFP28 for the same line rate. One example from our measurement is the 10G QSFP28 interfaces of the Wedge Switch with 0.31W, and the 10G RJ45 interface of the Cisco Nexus with 2.06W respectively.

Optimising energy consumption through port configuration

To get a better idea of the dynamic power and to compare it with its static counterpart, we use the model to estimate the dynamic power for a single port at maximum load at an arbitrarily selected frame size of 256 bytes. We assume the setup of the device to be the same as during the power model derivation and include the transceiver power portion to the interfaces. To derive the dynamic power, we use the following formula based on the power model:

$$P_{dyn}(c) = E_b(c) \cdot 2 \cdot r_{linerate}(c) + E_p(c) \cdot \frac{2 \cdot r_{linerate}(c)}{8 \cdot (256 + 20)} + P_{port,offset}(c) \tag{5.1}$$

where $r_{linerate}$(c) is the line rate depending on the interface configuration and is multiplied by 2 since we assume bi-directional traffic. A comparison of the static and dynamic power for a single port in this scenario is shown in Table 5.4.

DUT	c	$P_{port,sta}(c)$ [W]	$P_{dyn}(c)$ [W]
Wedge Switch	**QSFP28: 100G**	**1.57**	**1**
Wedge Switch	QSFP28: 25G	0.52	0.3
Wedge Switch	QSFP28: 10G	0.31	0.16
Cisco Nexus	**QSFP28: 100G**	**0.4**	**3**
Cisco Nexus	QSFP28: 40G	0.23	1.19
Cisco Nexus	RJ45: 10G	2.06	0.26
Cisco Nexus	RJ45: 1G	0.93	0.05
Cisco Catalyst	RJ45: 100M	0.21	0.01

Table 5.4: Interface power at maximum load with 256 bytes frame size

We observe in this traffic case that for almost all DUTs besides the QSFP28 interfaces of the Cisco Nexus, the dynamic power is smaller than its static power counterpart. In terms of port configuration, independent of the interface type, we observe that by lowering the line rate, the cost of the interface $P_{port,sta}(c)$ is usually lower. The same applies to the dynamic power: $P_{dyn}(c)$ decreases for low line rate configuration, which is mainly limited due to the maximum rate $r_{linerate}(c)$. To be more precise, we observe that while the energy cost per bit increases slightly for lower line rate configuration, the port processes significantly less bits due to the configured line rate. Hence, in total, the dynamic power decreases.

However, what is more interesting in this table are the two rows of the QSFP28 interfaces at 100G of the Wedge Switch and Cisco Nexus. We observe that the Wedge has a much higher interface cost than the Nexus. Contrary to that, the dynamic power of the Nexus in this scenario is three times as high as the Wedge. For simplicity, we do not consider the base power P_{base}. Thus, there is a trade-off between static and dynamic power of the port, and depending on the expected load, one port is more efficient than the other in terms of energy. This is shown in Fig. 5.1.

In this particular scenario, the 100G ports of the Cisco Nexus are more energy efficient than the Wedge Switch if the summed load is lower than 117 W, and vice-versa for larger traffic. Thus,

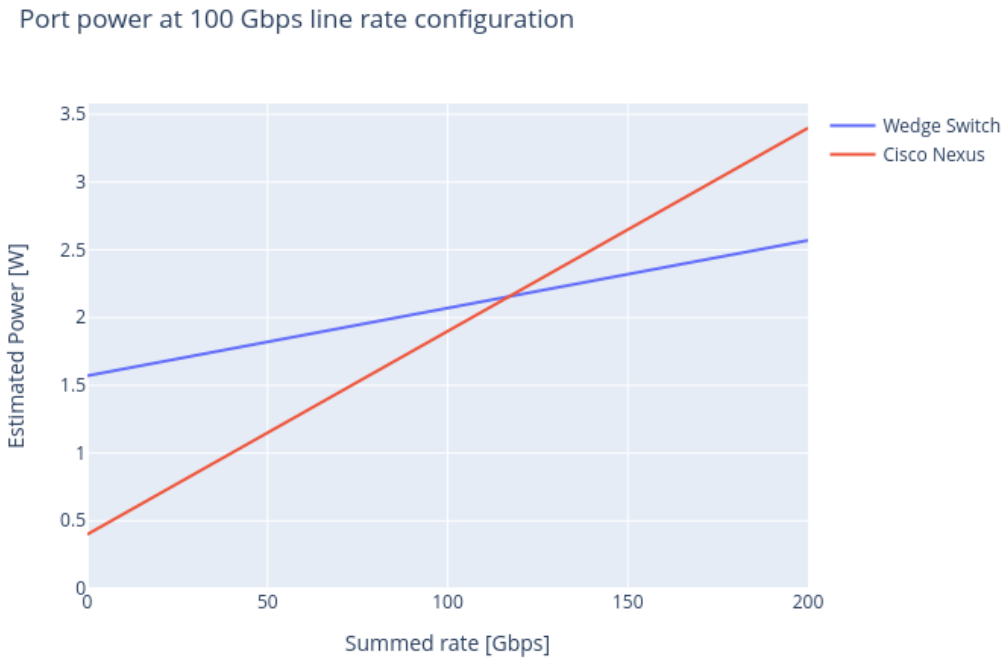


Figure 5.1: Total port power comparison

using the power model we can quickly identify trade-off of various switches in terms of performance and energy.

Frame size

In the methodology, we derived an equation that describes the total energy required to process a single frame of size L. Depending on the values of E_b and E_p, there is a certain threshold $L_{threshold}$ in which the cost of processing the frame is equal to the total energy cost of processing all bits of this frame. Based on Eq. (3.14), we determine this threshold value with:

$$8 \cdot L_{threshold}(c) \cdot E_b(c) + 8 \cdot 20 \cdot E_b(c) = E_p(c) \tag{5.2}$$

or equivalently:

$$L_{threshold}(c) = \frac{E_p(c) - 8 \cdot 20 \cdot E_b(c)}{8 \cdot E_b(c)} \tag{5.3}$$

For frames that are larger than $L_{threshold}$ bytes means more energy is required to process the bits of the frame, than processing the frame itself. We could also describe this in another way: Assuming we have a uniform distribution of frame sizes in the network, having a large $L_{threshold}$ value means that for most of the frames, the energy consumption for processing the frames is higher than the energy consumption of processing the bits and vice-versa.

Applying this equation to all our DUTs with their corresponding configuration, we obtain the frame size thresholds which are shown in Table 5.5.

What is interesting is that almost all threshold values are within the range of Ethernet frame sizes, that is, between 64 and 1518 bytes.

DUT	c	$L_{threshold}(c)$ [bytes]
Wedge Switch	QSFP28: 100G	504
Wedge Switch	QSFP28: 25G	257
Wedge Switch	QSFP28: 10G	199
Cisco Nexus	QSFP28: 100G	473
Cisco Nexus	QSFP28: 40G	317
Cisco Nexus	RJ45: 10G	287
Cisco Nexus	RJ45: 1G	47
Cisco Catalyst	RJ45: 100M	1515

Table 5.5: Frame size threshold for different configuration

Moreover, it seems that the lower the line rate, the lower the threshold value. This can also be seen by comparing the energy costs of each interface type in Tables 5.1 and 5.2: Within each interface type of the DUT, by lowering the line rate, the cost per bit increases, but at the same time, the cost per packet decreases. Hence, by Eq. (5.3), the threshold value decreases.

Plotting the data, we obtain Fig. 5.2. We observe that for the QSFP28 type ports, there is a linear trend of $L_{threshold}$ depending on the line rate configuration of the port. On the other hand, for the RJ45 no trend is visible as we do not have enough data, and moreover, the 100M interface from the Cisco Catalyst might have used different technologies since it is quite old. Thus, it seems for QSFP28 interfaces that the threshold increases linearly with the line rate configuration of the corresponding port. Or in other words: For lower line rate configuration, QSFP28 interfaces usually require more energy to process the bits of a frame than the frame itself.

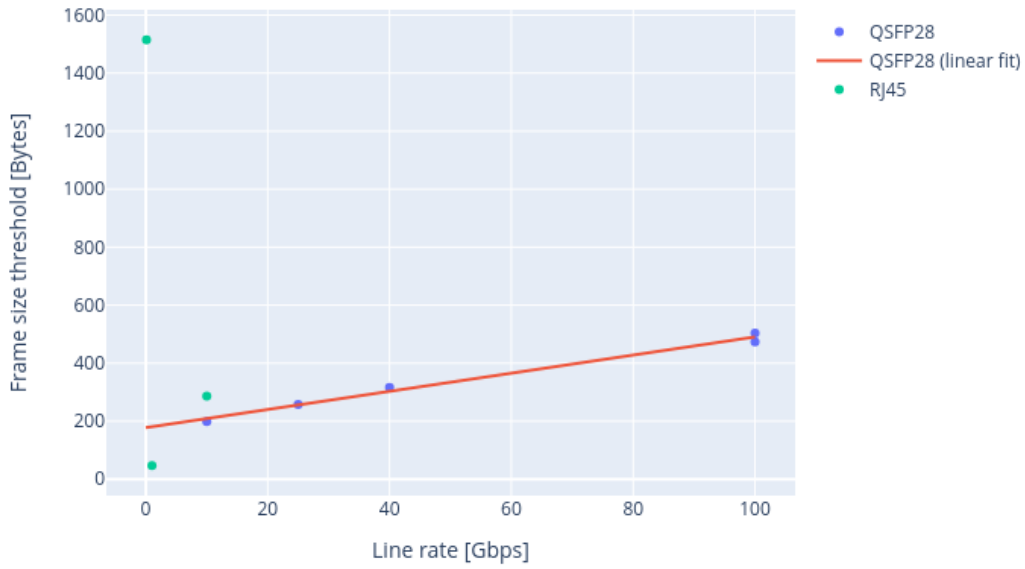


Figure 5.2: $L_{threshold}$ depending on the line rate configuration

In terms of energy optimisation, it should be noted that the threshold value $L_{threshold}$ is a metric that describes E_b and E_p relative to each other for a given configuration. Of course, we can reduce the overall dynamic power by processing as few packets and bits as possible. However, let us assume that we can choose the length of a given frame. For a port with a low frame threshold value, it is better to process a smaller frame as the cost of E_b is high, relative to E_p. On the other hand, for a port with a large frame threshold value, it is more efficient to process a larger frame because the additional cost per bit is relatively low to the packet process cost. Hence, it makes sense to send more bits for a relatively low bit energy cost.

Concrete comparison of energy efficiency

To determine which of our DUTs is the most energy efficient device for example, we use a simple scenario and assume that 6 ports are on average utilized at 20G (summed rate!) with frame sizes of 256 bytes. Using our model, the expected power at this particular load is given by:

$$P_{switch} = P_{base} + 6 \cdot (P_{port,sta}(c) + E_b(c) \cdot 20 \cdot 10^9 + E_p(c) \cdot \frac{20 \cdot 10^9}{8 \cdot (256 + 20)} + P_{port,offset}(c)) \quad (5.4)$$

Assuming we configure our DUTs in such a way that we only activate 6 ports and configure them equally, we obtain the following expected power sorted by the total switch power P_{switch}:

DUT	c	P_{switch} [W]	P_{sta} [W]	P_{dyn} [W]	Difference to least efficient config [W]
Wedge Switch	QSFP28: 10G	110.89	109.96	0.93	7.23
Wedge Switch	QSFP28: 25G	112.13	111.22	0.91	5.99
Wedge Switch	QSFP28: 100G	118.12	117.52	0.6	0
Cisco Nexus	QSFP28: 40G	150.31	148.39	1.92	10.62
Cisco Nexus	QSFP28: 100G	151.21	149.41	1.80	9.72
Cisco Nexus	RJ45: 10G	160.93	159.37	1.56	0

Table 5.6: Total power usage depending on port configuration

Comparing the configuration of the devices, lowering the line rate of the interfaces on both devices is the optimal setting for energy efficiency. Though this is not the case for the RJ45 ports on the Cisco Nexus: The interface cost of the RJ45 port is significantly higher than the QSFP28 port. We note that while the dynamic power slightly increases for lower line rate settings on the QSFP28 interfaces, the static power cost has a greater impact on the total power and reduces it overall.

In this particular scenario, if we lower the line rate of the Wedge Switch from 100G to 10G, we reduce the total switch power by 6.1%. Similarly, if we use the 40G QSFP28 ports instead of the 10G RJ45 ports, we would lower the switch power by 6.6%.[1].

Wedge Switch: Low energy costs

One observation is that on the Wedge Switch, the energy costs are noticeably lower than on the Cisco Nexus for the QSFP28 interfaces. We believe this may be the case because the data plane

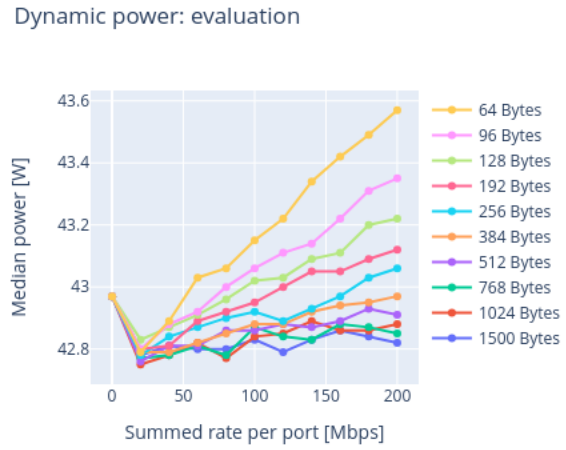
[1]We are aware that the QSFP28 interfaces of the Cisco Nexus are usually reserved for uplink connections which require much higher bandwidth and are therefore not suitable for such a scenario. However, we include them here for the sake of comparison.

program on the Wedge is programmed in a very simple way: We forward the packets along the snake route depending on which ingress port they arrived at, and therefore independent of the MAC addresses. From our previous work, we know that the data plane of the Wedge Switch does indeed affect the total power. Thus, if we were to program the data plane with basic L2 functionality, and we then repeat the snake test with the new data plane, it is very likely that the energy costs would increase, however, only slightly as the functionality of L2 switches does not require very complex logic. Therefore, in the previous comparison of the energy efficiency between the Wedge Switch and the Cisco Nexus, we would not observe any significant changes for a relatively small load (20G at 6 ports each).

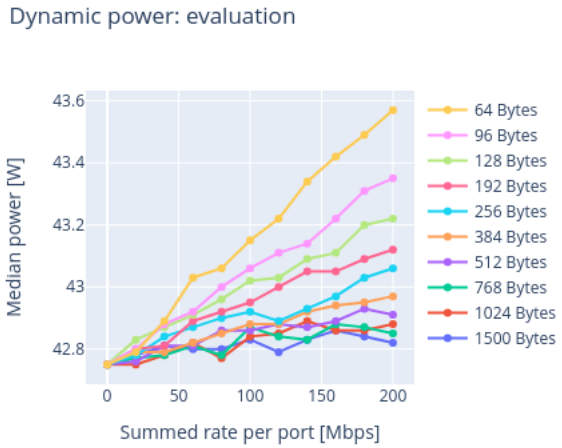
Power offset

For almost every DUT, we observe a power offset $P_{port,offset}(c)$ that appears if low traffic is being processed, contrary to the power at 0 Gbps. As a matter of fact, for all QSFP28 configurations, except for the 100G line rate, the offset is positive. We believe this offset comes from a mechanism at the switch where the interface switches into a low-power state if no traffic is detected for a longer time period. Similar ideas are discussed in the Energy Efficient Ethernet [19]. It is also possible that this offset comes from the transceiver rather than the switch.

Moreover, this offset is even negative for all RJ45 interfaces: Fig. 5.3 shows the difference in including the offset to the derivation of the coefficient a_L on the Cisco Catalyst (RJ45 interface with 100M line rate).



(a) At 0 Mbps, no traffic is sent.



(b) At 0 Mbps, low bi-directional traffic (1 Mbps) is effectively sent.

Figure 5.3: Derivation of a_L on Cisco Catalyst for 100M line rate

We observe that, when no traffic is sent (Fig. 5.3a), the power is even higher than in other cases where the DUT is utilized and does not match our assumption that the power scales linearly with the rate. On the other hand, in the low traffic scenario in Fig. 5.3b, the power fits much better to the linear behaviour. Therefore, we replace the data point at 0 Mbps with the low traffic scenario and add the offset term in the model to include the power difference. Moreover, on the Cisco Catalyst in the low traffic scenario, we send 1 Mbps traffic with 1500 bytes frame size, since at 1 pps (or 12.1 kbps), we observed no difference in power compared to the no-traffic scenario.

However, it is unclear to us where this negative offset comes from. One speculation is that this behaviour is device-specific and may only occur on Cisco devices, and therefore depends on the circuit design of the device.

Cisco Nexus: Power behaviour

We observed a power pattern on the Cisco Nexus switch where the power draw spikes for about 15 seconds in any case. This pattern repeats every 76 seconds. Therefore we set the measurement time for the Cisco Nexus to match the interval to avoid any influence of the pattern on our power data. Fig. 5.4 shows this power behaviour. However, we do not understand the source of this behaviour and have contacted Cisco about this observation for further investigation.

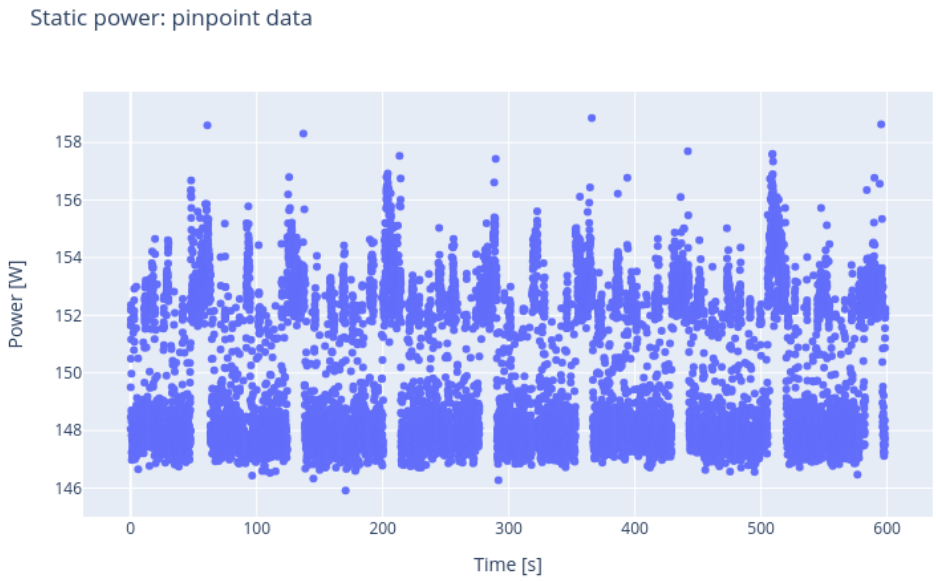


Figure 5.4: Power measurement of the Cisco Nexus. All ports are shut down.

We also observed on the Cisco Nexus that when two additional AOC transceivers (of type LR, and LR4) were connected to another device, the power increased by 3.9 W even while both interfaces were inactive. From this observation, we added the additional term $P_{trx,plugged}$ to account for this increase in power, which we assume must come from the transceiver. On the other hand, we did not observe such behaviour on the Wedge Switch. Though again, this depends on the circuit design of the device.

5.2 Validation

To validate the power model, we additionally implement a traffic script that varies the traffic load over time in the snake setup. The problem with this validation process is that we use the exact setup from which we derived our model parameters. This is because, with our current testbed featuring one server, we cannot generate sufficient traffic to utilize the DUT more realistically, i.e. random traffic load over multiple ports at a time.

We are currently investigating the topic of validation by measuring the power of a Wedge Switch deployed by SWITCH [16] at the University of Zurich and validating our power model with the measured data.

In particular, we are measuring the power of the switch over longer periods of time with an external power meter MCP39F511N we used in our power modelling. In addition, we obtained traffic data (specifically, the number of packets and bytes processed on several interfaces every 5 minutes) from SWITCH from which we estimate how much traffic is being processed on average over the ports.

The deployed switch has 7 active interfaces, each equipped with optical transceiver modules that differ from ours during the power modelling process. Since optical transceivers typically require more power, our model parameters for the electrical transceiver power do not fit this setup. Therefore, we replace the term P_{trx} with the maximum power usage of the transceivers based on their datasheet. Moreover, we use the power model including the parameters of the Wedge Switch (and the transceiver power from the datasheets) to estimate the effective power usage.

At the time of writing, we have measured the switch power over 24 hours. We set our measurement sampling interval to 5 seconds and averaged 60 data points together so that the number of data points from the measurement and estimate match (concretely, 1 data point every 5 minutes).

By comparing the measured data and the estimated power from the model, we obtain the following figure:

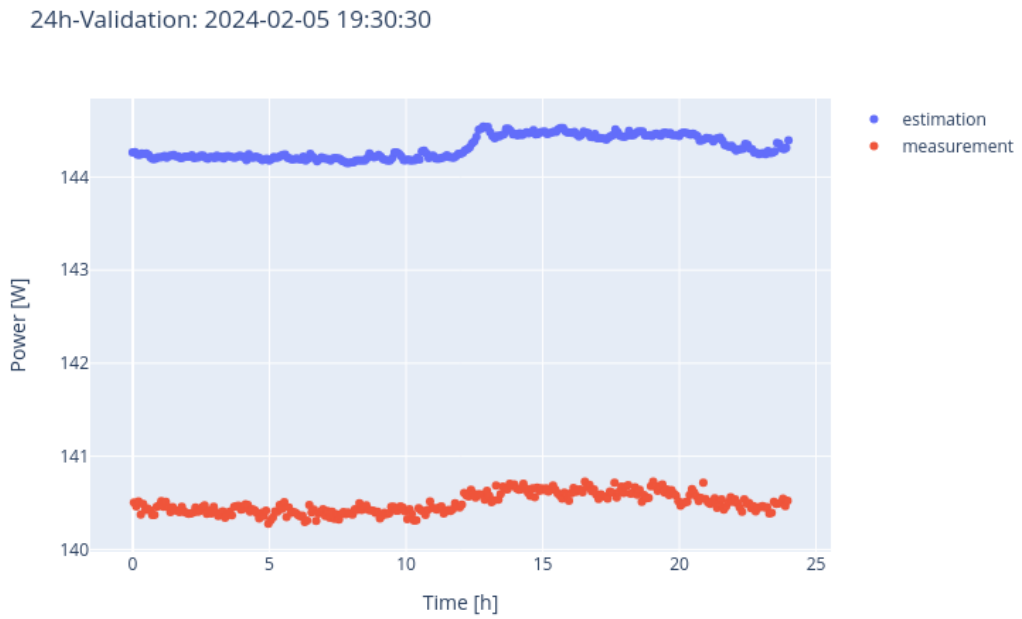


Figure 5.5: Power measurement and estimation of deployed Wedge Switch.

We observe that the estimated power is roughly 3.8W higher than the measured data. This may be due to the transceivers: We have included the *maximum* power cost based on their specification. The real cost may be lower. On the other hand, we note that there are 4 interfaces on the switch that appear to handle small loads, but we do not have the corresponding traffic data for these ports. Therefore a small portion of the dynamic power from these interfaces is missing in the estimate.

There are errors in our estimation due to missing information. However, if we subtract the 3.8W difference as a constant in our power model, we obtain Fig. 5.6.

We observe that by including the constant offset, the power model seems to give a good estimate of the effective power draw of the deployed device.

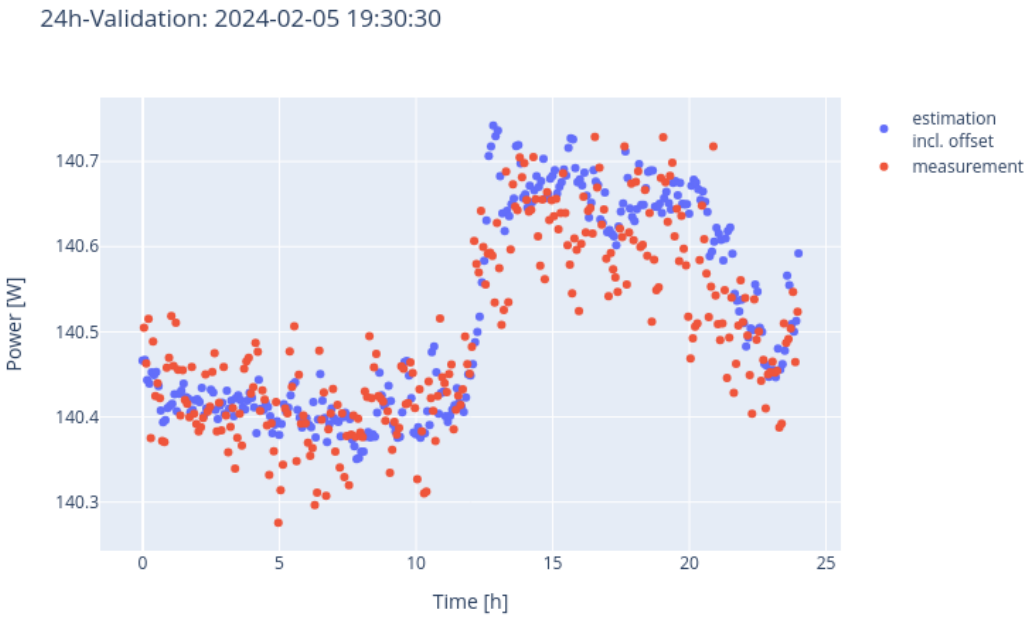


Figure 5.6: Power measurement and estimation (incl. -3.8W offset) of deployed Wedge Switch.

Currently, we are continuing to investigate the validation of power usage by measuring the power of the deployed switch over longer periods such as weeks. However, from what we have seen so far, apart from a constant offset that can be explained by missing information, the power model seems to give a good estimate of the actual power draw of the device.

Chapter 6

Outlook

We described a framework for robustly evaluating a power model for L2 switches. Such a framework allows us to get a better understanding of the energy consumption of switches. Specifically, modelling the power usage allows us to understand which component contributes how much to the total power. And by collecting the data from a larger set of devices, we can observe certain trends related to the energy consumption.

In addition, our model focuses on the configuration of the device, mainly the interface setting. This allows us to compare the energy consumption of different port configurations over multiple devices, and it gives us an insight into how much energy we can potentially save by configuring our switches appropriately.

The framework is designed so that it can be applied to various network switches. By collecting power data from a larger set of switches, we can perform accurate power analysis at the network level. This helps us answer questions like *how should we select/configure our devices in the network to reduce the overall power usage while maintaining a reasonable performance?* and encourages sustainable practices.

In terms of future work, we discuss some of the related topics in the following:

Competitive energy benchmark

While it is a valid approach to compare the power draw of network devices through an energy benchmark, it does not provide deeper insight into the energy consumption of the device component.

The snake test is one potential profile that an energy benchmark could use. Unlike our methodology for determining the cost of static power parameters which is very restrictive, the snake test can be specified to be less confining and allows a benchmark user to configure to be as energy-efficient as possible, for example by allowing the data plane and the interfaces of the DUT to be configured to be more optimized for energy. Designing such benchmarks can help network operators select network equipment optimized for power and can also incentivize vendors to develop more energy-efficient equipment.

Extending the power model

The power model can be extended to include other characteristics. Examples we described earlier are the power supply unit efficiency and the fan power.

Another suggestion is to extend the model to L3 devices. However, this may need to include more complex factors such as the impact of the control plane. This can be challenging if we try to analyze it through external power measurements as in our current setup.

Moreover, the transceiver power model is not fully understood. In our work, we assumed that their power component is constant. However, we do not know for sure whether this is true. It may be that the power usage scales with the traffic load, therefore, the characterization of the power of various transceiver types is left for further investigation.

Design of a public database

The long-term vision of this project is to collect power data from many different network devices. We envision that this data will be shared publicly. Therefore, the design of a public database is beneficial for collecting the shared data.

Bibliography

[1] 100G QSFP28 AOC with CDR (TX / RX) | Multimode Fiber, 0.5 - 100 m. https://www.flexoptix.net/de/q-a851hg-z.html.

[2] 100G QSFP28 DAC | Passive Copper Cable, 0.5 - 5 m. https://www.flexoptix.net/de/q-czz1hg-z.html.

[3] Benchmarking Methodology for LAN Switching Devices. Tech. rep.

[4] CISCO CATALYST 3560 Series Switches Hardware View. https://www.cisco.com/web/ANZ/cpp/refguide/hview/switch/3560.html#ws-c3560-24ps.

[5] Cisco Nexus 93108TC-FX Switch. https://www.cisco.com/c/en/us/support/switches/nexus-93108tc-fx-switch/model.html.

[6] Cisco TRex. https://trex-tgn.cisco.com/.

[7] DCS800: Wedge 100BF-32X. https://www.edge-core.com/product/dcs800/.

[8] DPDK. https://www.dpdk.org/.

[9] Intel® Tofino™. https://www.intel.com/content/www/de/de/products/details/network-io/intelligent-fabric-processors/tofino.html.

[10] L.1310 : Energy efficiency metrics and measurement methods for telecommunication equipment. https://www.itu.int/rec/T-REC-L.1310-202009-I/en.

[11] MCP39F511N POWER MONITOR DEMONSTRATION BOARD. https://www.microchip.com/en-us/development-tool/adm00706.

[12] Measurement methods for energy efficiency of router and switch equipment. https://www.etsi.org/deliver/etsi_es/203100_203199/203136/01.02.01_60/es_203136v010201p.pdf.

[13] Network and Telecom Equipment - Energy and Performance Assessment. https://www.itu.int/md/T09-FG.ICT-C-0072/en.

[14] NVIDIA Mellanox ConnectX-5 adapters. https://www.nvidia.com/en-us/networking/ethernet/connectx-5/.

[15] P4. https://opennetworking.org/p4/.

[16] Switch. https://www.switch.ch/de.

[17] Transmittal of ATIS Energy Efficiencies Specifications. https://www.itu.int/md/T09-FG.ICT-IL-0003/en.

[18] Benchmarking Methodology for Network Interconnect Devices. Request for Comments RFC 2544, Internet Engineering Task Force, Mar. 1999. Num Pages: 31.

[19] Energy Efficient Ethernet. https://en.wikipedia.org/wiki/Energy-Efficient_Ethernet, Mar. 2021.

[20] 80 Plus. https://en.wikipedia.org/w/index.php?title=80_Plus, Jan. 2024.

[21] Ethernet frame. https://en.wikipedia.org/w/index.php?title=Ethernet_frame, Jan. 2024. Page Version ID: 1193320467.

[22] nsg-ethz/power-bench. https://github.com/nsg-ethz/power-bench, Feb. 2024. original-date: 2024-02-05T20:53:22Z.

[23] osmhpi/pinpoint. https://github.com/osmhpi/pinpoint, Jan. 2024. original-date: 2020-07-30T16:14:44Z.

[24] Ahn, J., and Park, H.-S. Measurement and modeling the power consumption of router interface. In *16th International Conference on Advanced Communication Technology* (Feb. 2014), pp. 860–863. ISSN: 1738-9445.

[25] Almes, G. T., Mahdavi, J., Mathis, M., and Paxson, V. Framework for IP Performance Metrics. Request for Comments RFC 2330, Internet Engineering Task Force, May 1998. Num Pages: 40.

[26] Avramov, L., and jhrapp@gmail.com. Data Center Benchmarking Methodology. Request for Comments RFC 8239, Internet Engineering Task Force, Aug. 2017. Num Pages: 19.

[27] Bianzino, A. P., Raju, A. K., and Rossi, D. Apples-to-apples: a framework analysis for energy-efficiency in networks. *ACM SIGMETRICS Performance Evaluation Review 38*, 3 (Jan. 2011), 81–85.

[28] Dugatkin, D., Hamza, A., Velde, G. V. d., and Popoviciu, C. IPv6 Benchmarking Methodology for Network Interconnect Devices. Request for Comments RFC 5180, Internet Engineering Task Force, May 2008. Num Pages: 20.

[29] Guo, F., Ormond, O., Collier, M., and Wang, X. Power measurement of NetFPGA based router. In *2012 IEEE Online Conference on Green Communications (GreenCom)* (Sept. 2012), pp. 116–119.

[30] Hickman, B., and Stopp, D. J. Methodology for IP Multicast Benchmarking. Request for Comments RFC 3918, Internet Engineering Task Force, Oct. 2004. Num Pages: 31.

[31] Lim, J. How much does it burn? Profiling the energy model of a Tofino switch. Accepted: 2023-06-27T10:04:55Z Publisher: ETH Zurich.

[32] Lindblad, J., Mitrovic, S., Palmero, M., and Salgueiro, G. Power and Energy Efficiency. Internet Draft draft-opsawg-poweff-00, Internet Engineering Task Force, Oct. 2023. Num Pages: 37.

[33] Mahadevan, P., Sharma, P., Banerjee, S., and Ranganathan, P. A Power Benchmarking Framework for Network Devices. In *NETWORKING 2009* (Berlin, Heidelberg, 2009), L. Fratta, H. Schulzrinne, Y. Takahashi, and O. Spaniol, Eds., Lecture Notes in Computer Science, Springer, pp. 795–808.

[34] Manral, V., Sharma, P., Banerjee, S., and Ping, Y. Benchmarking Power usage of networking devices. Internet Draft draft-manral-bmwg-power-usage-04, Internet Engineering Task Force, Mar. 2013. Num Pages: 16.

[35] Rodoni, L. High-speed Traffic Generation.

[36] 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)* (Apr. 2011), pp. 331–336.

[37] Vishwanath, A., Zhi Zhao, Sivaraman, V., and Russell, C. An empirical model of power consumption in the NetFPGA Gigabit router. In *2010 IEEE 4th International Symposium on Advanced Networks and Telecommunication Systems* (Mumbai, India, Dec. 2010), IEEE, pp. 16–18.

[38] Vishwanath, A., Zhu, J., Hinton, K., Ayre, R., and Tucker, R. S. Estimating the energy consumption for packet processing, storage and switching in optical-IP routers. In *2013 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)* (Mar. 2013), pp. 1–3.

Appendix A

My Appendix

A.1 Example: Dynamic power evaluation from Cisco Catalyst

In this section, we present one result of the dynamic power evaluation of the Cisco Catalyst using 10G line rate configuration on the RJ45 interfaces.

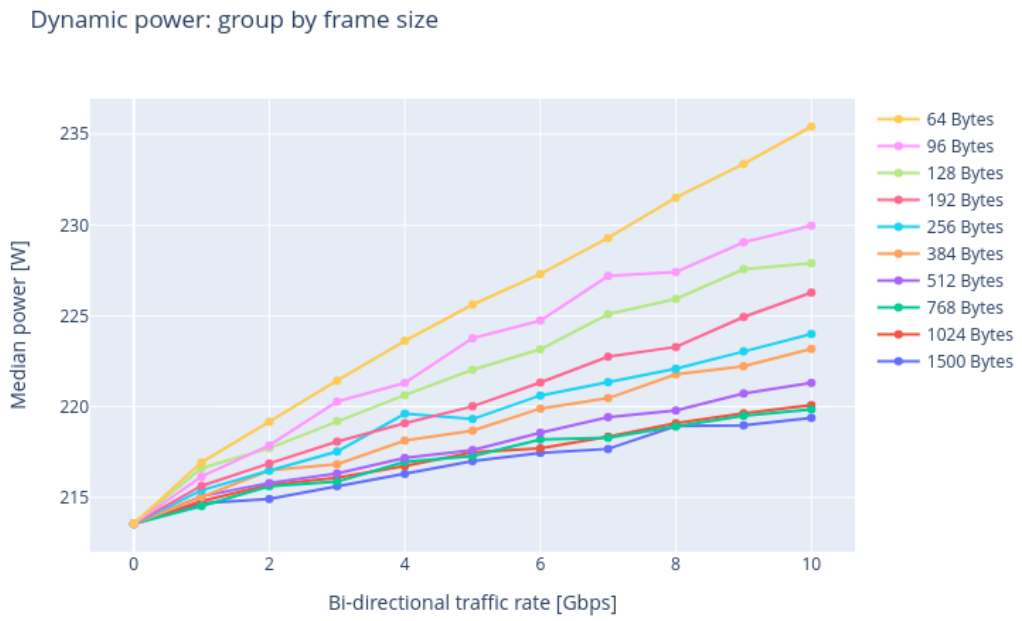


Figure A.1: Dynamic power evaluation: We observe a linear increase in power depending on the rate. The smaller the frame size L, the higher the corresponding slope a_L. We derive slopes a_L for each frame size L through linear regression.

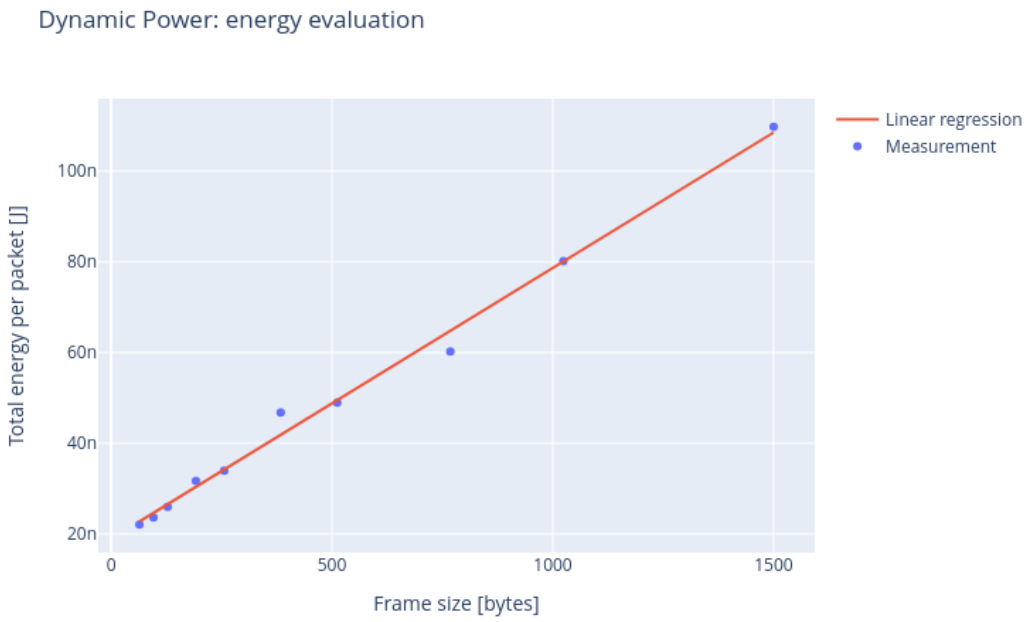


Figure A.2: Total energy cost for processing a frame depending on frame size L. Each data point corresponds to the right hand side of Eq. (3.14): $8 \cdot (L + 20) \cdot a_L$. By estimating slope and intercept with linear regression, we can determine the energy cost per bit and the energy cost per packet.