Fantastic Joules and Where to Find Them. Modeling and Optimizing Router Energy Demand

Conference Paper

Author(s):
Jacob, Romain; Röllin, Lukas; Lim, Jackie; Chung, Jonathan; Béhanzin, Maurice; Wang, Weiran; Hunziker, Andreas; Moroianu, Theodor; Tabaeiaghdaei, Seyedali; Perrig, Adrian; Vanbever, Laurent

Publication date:
2025-10

Permanent link:
https://doi.org/10.3929/ethz-b-000728960

Originally published in:
https://doi.org/10.1145/3730567.3732920

Fantastic Joules and Where to Find Them. Modeling and Optimizing Router Energy Demand

Romain Jacob*
Lukas Röllin
jacobr@ethz.ch
ETH Zürich
Zürich, Switzerland

Jackie Lim
Switch
Zürich, Switzerland

Jonathan Chung
Maurice Béhanzin
Weiran Wang
Andreas Hunziker
ETH Zürich
Zürich, Switzerland

Theodor Moroianu
Seyedali Tabaeiaghdaei
Adrian Perrig
Laurent Vanbever
ETH Zürich
Zürich, Switzerland

Abstract

Reducing our society's energy demand is critical to address the sustainability challenge. While the Internet currently accounts for 1—1.5% of global electricity consumption and continues to grow, the energy demands of one of its core components—routers—remain poorly understood. The available power data is limited and not fine-grained enough, offering little actionable insight into strategies for effectively reducing the Internet's energy consumption.

To address this, we assemble and present a unique dataset including datasheet information, router-internal measurements, external power measurements, and router power models. This dataset depicts a clearer picture of routers' energy demand and provides insights on how to reduce it. Our initial analysis of the dataset suggests, *e.g.*, that (i) datasheets are not useful predictors, sometimes even incorrect; (ii) internal router power measurements have limited accuracy; (iii) using more efficient and better-sized power supply units is a promising energy-saving vector; (iv) turning links off is less efficient than anticipated in the literature. This work also highlights the limitations of today's power monitoring practices and provides suggestions for improvement.

CCS Concepts

• **Networks → Network performance modeling**; *Network performance analysis.*

Keywords

Power models, Power optimization

ACM Reference Format:
Romain Jacob, Lukas Röllin, Jackie Lim, Jonathan Chung, Maurice Béhanzin, Weiran Wang, Andreas Hunziker, Theodor Moroianu, Seyedali Tabaeiaghdaei, Adrian Perrig, and Laurent Vanbever. 2025. Fantastic Joules and Where to Find Them. Modeling and Optimizing Router Energy Demand. In *Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25), October 28–31, 2025, Madison, WI, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3730567.3732920

*A CRediT statement for this work is available in § A.

ACM ISBN 979-8-4007-1860-1/2025/10
https://doi.org/10.1145/3730567.3732920

1 Introduction

The Information and Communication Technology (ICT) sector was estimated to consume 4% of the global electricity used in 2020 [24], a large portion of which is attributed to the Internet infrastructure (1-1.5% in 2022 [19], of which mobile accounts for ≈ 2/3). To reduce the Internet's energy footprint, we must understand what drives its energy demand. Perhaps surprisingly, while hardware vendors increasingly market their sustainability efforts and efficiency improvements (*e.g.*, Fig. 2a), we lack a detailed understanding of the power demand of the Internet and of routers[1] in particular.

We start by examining currently available router power data. The first source is datasheets, where hardware vendors typically provide only a few broad metrics, such as "maximum power" and "typical power," sometimes accompanied by specific operating conditions such as "at 25°C" or "at 1.8 Tbps." These numbers are primarily intended for operators to dimension the power supply of a rack or server room. Thus, they are expected to overestimate the actual power drawn by a router in deployment. Sometimes, though, datasheets are not only imprecise; they can be incorrect (§ 3).

The next data source comes from measuring the power demand of routers in deployment. Nowadays, most power supply units (PSU) measure the power they deliver, which can be exported and collected via standardized MIB objects in SNMP or similar. It is unclear, however, whether those measurements are trustworthy.

The last source of information comes from power models. While such models have been proposed in the past [35], they were never validated on production networks.

Research questions. This paper studies various power data sources to better understand the energy demand of routers. More specifically, we investigate the following questions:

Q1 Are datasheets representative of router power in practice? Do they confirm the power efficiency improvements claimed by vendors?
Q2 Are the PSU power measurements trustworthy?
Q3 Can we precisely predict router power in the wild?

Addressing **Q1** requires analyzing highly unstructured and irregular datasheets. Addressing **Q2** requires measuring router power draw externally *and* collecting PSU-reported numbers by the same device. Today, some ISPs routinely monitor router power information, but few agree to share that data, and even fewer are willing to provide

[1] In this paper, we use "router" to denote "wired network devices," including switches but excluding wireless access points and home routers.

physical access to their devices to perform external power measurements. Finally, answering **Q3** requires measuring the power drawn by production routers while collecting real-time traffic statistics.

Dataset. To investigate those questions, we collect, curate, and present a unique dataset of router power data, including:

- a collection of datasheet power values spanning 777 router models from Cisco, Juniper, and Arista;
- 10-month-long traces of PSU measurements and interface traffic counters collected via SNMP from 107 routers deployed by Switch, a Tier-2 ISP;
- 2-month-long traces of external power measurements of three routers deployed by Switch;
- fine-grained power models for eight router models.

Tools. The collection and curation of this dataset are supported by several tools that are all publicly available.

Datasheet parser A scalable parser leveraging LLMs to extract information from unstructured datasheets (§ 3.2);

Autopower A system to facilitate external power measurements of routers in production (§ 6.1);

NetPowerBench A complete set of tools to orchestrate the derivation of router power models in the lab, including traffic generation and processing of the measurement data (§ 5);

Network Power Zoo A public database to aggregate all types of network power data [18], open for the community to use and contribute to.

Main findings. We analyze our dataset to provide some preliminary answers to our research questions.

§ 3 As expected, datasheets do not provide reliable estimates of the actual power demand nor do they clearly show energy efficiency improvements over time.

§ 6 PSU measurements are inconsistent. On some routers, they appear precise albeit not accurate (*i.e.*, there is a constant offset to the true value); on others, they are pseudo-constant with little information.

§ 6 The power model first proposed by [35] can precisely predict (though with an offset) router power draw. We validate this for the first time in the wild.

Empowered by this enhanced understanding of router power behavior, we explore several energy-saving approaches.

§ 7 Consistent with the literature [26], we confirm that the amount of traffic carried by a router only slightly increases its power consumption. We illustrate this in Fig. 1 which depicts the total power and traffic volume for Switch over time.[2] In particular, we note that the correlation between power and traffic is invisible at the network scale.

§ 7 We also confirm that optical transceivers account for a sizable part of a router's total power. They represent ≈ 10% in the Switch network. Thus, aiming to optimize transceiver power demand seems promising.

§ 7 However, it appears that turning a port "down" does not always power off transceivers; a large part of a transceiver's power cost appears as soon as plugged in. That does not seem to be a fundamental limitation and may be more a software problem than a hardware one, thus possibly easy(ier) to address.

§ 8 Consequently, turning off links as suggested in [31] is expected to yield smaller power savings than anticipated, between 0.4–1.9% in the Switch network.

§ 9 The offset introduced by the power model may come, in part, from varying power conversion efficiencies across PSUs of the same router model. We start exploring several means of improving PSU efficiencies and estimate one could save up to 9% in the Switch network *without* any impact on the routing state or network configuration. A deeper investigation of PSU efficiency is necessary, but it appears promising.

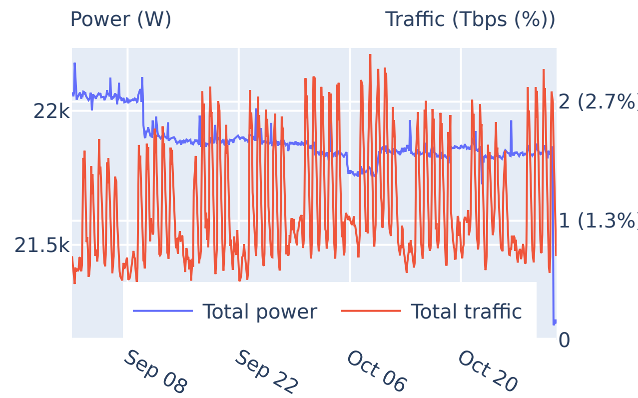


Figure 1: Total power draw and traffic volume from all routers in the network of Switch, a Tier-2 ISP.

We conclude by discussing lessons learned and suggesting important next steps. This paper focuses on providing some foundations and tooling for future sustainable networking research. Our analysis for energy-saving approaches (§ 7 to 9) is preliminary and requires additional research.

All the data and software required to replicate the analyses presented in this paper are publicly available [20].

2 Related work

In 2003, Gupta and Singh [17] raised awareness on the Internet energy usage in the networking community and triggered a generation of research on the topic. That line of work revolves around the network state—*e.g.*, which links, linecards, or routers can be turned off — but struggles to relate those results to power savings. After Nedevschi et al. [27] laid out the theoretical foundation of energy savings in wired networks, numerous works (*e.g.*, [22, 23, 34, 37]) studied the question further but had to resort to using the number of links or linecards as a proxy for network power; ultimately, these works abstract away network power optimization as a traffic engineering problem. Otten et al. [29] question that approach and find that linecard power dominates for the routers they consider, thus the number of links turned off is a poor proxy for energy savings.

Using counts of links or linecards as a proxy for power is not sufficient because real networks are heterogeneous: the power draw of a network link depends, *e.g.*, on the router type, the port type, the

[2]Power changes generally coincide with hardware (de)commissioning.

transceiver model, and the traffic volume. One needs a more refined power model to assess network energy-saving schemes better. Some recent works [16, 33] made one step in that direction by modeling router power based on datasheet information, based on reported idle power, max power, and maximum throughput. While this still lacks granularity (*e.g.*, there is no account of the transceiver power) and datasheet information may be unreliable (§ 3), it better captures the heterogeneity of a real network.

The work from Vishwanath et al. [35] is the first, to our knowledge, to tackle router power modeling. Through a set of carefully crafted experiments, the power draw of different router elements can be estimated via a power-delta approach. Our modeling approach is very similar (see § 5 for details). The main shortcoming of [35] is that the accuracy of the power models was not validated in the real world. Moreover, the power modeling software was never released. With this paper, we address both those limitations: we re-implement and release a router power modeling framework [32] and validate that the methodology indeed produces accurate power models, which can be meaningfully used in downstream network power optimization research (*e.g.*, § 8 and 9). Osa Mostazo et al. [14] conduct a parallel effort with an approach similar to ours; they derive similar power models and focus on the hardware deployed in the Telefonica network.

3 Router datasheet analysis

We first consider the only publicly available source of power data: datasheets. Specifically, we aim to assess whether energy efficiency improvements over time are visible in the datasheets and whether datasheet power values represent the router power draw measured in practice (**Q1**).

3.1 We thought it would be easy...

To study the trends, we aim to automate the datasheet parsing to scale the collection to a large number of models. More specifically, our objective is to collect:

- Typical and max power draw;
- Number and capacity of power supply units;
- Maximum bandwidth;
- Release, end-of-sale, and end-of-support dates.

Since router datasheets are publicly accessible online, aggregating the relevant power information appears straightforward. In practice, however, several challenges arise.

(1) One needs a sensible list of relevant router models to search datasheets of.
(2) Often, router models belong to "series," *e.g.*, the Cisco 8000 series, with only one datasheet for all the models in the series, but the mapping between router models and series is not always obvious. Furthermore, release dates are typically published for an entire series (not for the individual router model) and are often not included in the datasheets.
(3) More importantly, the datasheet information is unstructured and irregular. First, the same data, *e.g.*, maximum bandwidth, comes under many different names, even for the same vendor. In some cases, the maximum bandwidth is not directly written and must be derived by summing the ports' capacities. Moreover, routers sometimes come with different options that are not always reflected in the model name; *e.g.*, some router models offer different power supply unit capacities, resulting in multiple power numbers. Sometimes, power information is simply absent, or even kindly reported as "TBD" [1]. Finally, the actual presentation of the data varies widely. The relevant numbers may be mentioned in the middle of paragraphs or burried somewhere in a large table spanning several pages of obscure multi-column and multi-row layouts.

3.2 ... so we did it anyway

We solve the first problem mentioned above by starting from an existing device inventory for the NetBox network management software [10]. It provides a structured collection of device models in YAML format organized by vendors, which includes a field with datasheet URLs. The number and capacity of PSUs is also collected from NetBox if present.

Based on that collection of URLs, we use the `GPT-4o` [4] large language model (LLM) to automate the data extraction of power and bandwidth values, as well as inferring the router model series. However, it proved unable to return accurate release date information; all the release dates present in our dataset have been collected manually. Manual verification of sampled outputs showed that the results are reasonably accurate but—as one would expect—far from perfect. Our dataset identifies the LLM outputs (subject to "hallucinations") from the other data sources, either manually collected or imported from NetBox.

> All the software used to extract the datasheet information, including the LLM prompts, is publicly available [36]. The resulting collection of datasheet information is part of the Network Power Zoo [18].

3.3 Datasheet analysis

Using the methodology described in § 3.2, we extract datasheet information from a total of 777 router models from Cisco, Arista, and Juniper.[3] At the time of writing, the dataset contains release dates for Cisco devices only, as we were yet unable to scale the data collection for the other vendors.

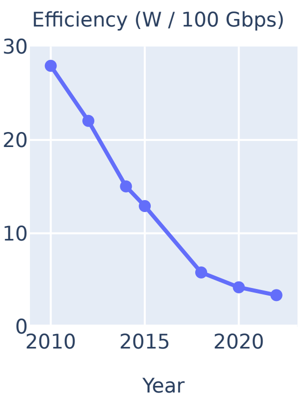
3.3.1 Do datasheets reflect power efficiency improvements over time? Marketing discourse is adamant: "Technology continuously improves, and new hardware is more efficient than the old." For example, Fig. 2a shows the efficiency improvements of Broadcom's switching ASICs over time, as reported by Broadcom. But even if individual components improve, it does not necessarily mean the overall system does. We thus investigate whether the datasheet numbers actually show power efficiency numbers over time.

Method. From the datasheets present in our dataset, we compute the same power efficiency metric used in Fig. 2a: typical power per 100 Gbps of traffic. We used the maximum power number whenever datasheets do not report values for typical power. Finally,

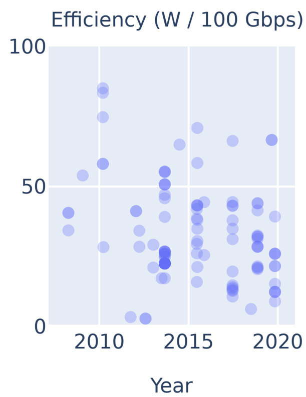
[3]The choice of vendors is arbitrary.

Table 1: The "typical" power reported by datasheet says little about the actual power draw. Surprisingly, some datasheets even *underestimate*.

Power values / Router model	Measured *Median*	Datasheet *"Typical"*	
NCS-55A1-24H	358 W	600 W	40 %
ASR-920-24SZ-M	73 W	110 W	33 %
NCS-55A1-24Q6H-SS	285 W	400 W	28 %
NCS-55A1-48Q6H	346 W	460 W	24 %
ASR-9001	335 W	425 W	21 %
N540-24Z8Q2C-M	159 W	200 W	20 %
8201-32FH	359 W	288 W	-24 %
8201-24H8FH	296 W	205 W	-44 %



(a) Power efficiency trend of Broadcom's switching ASICs. Redrawn from [21].



(b) Power efficiency trend from the datasheet numbers present in our dataset.

Figure 2: The efficiency improvement trend, clearly visible at the ASIC level (Fig. 2a), is not as obvious from the router datasheet numbers (Fig. 2b).

we only consider routers with a capacity larger than 100 Gbps, as the efficiency metric is intended for "high-end" routers; the metric yields very large numbers for smaller access points.

Result. Fig. 2b shows the evolution over time of the power efficiency as reported by the datasheets. Two routers released in 2008 and 2011 are not shown due to outlier values around 300, which would hinder the plot's readability. The trend in total system efficiency (Fig. 2b) is not as clear as for ASIC efficiency (Fig. 2a); additional data is required to draw a strong conclusion. In any case, while technological improvements in individual components are helpful, they are not sufficient to drive system-wide improvements. Naturally, that conclusion holds assuming that datasheet power numbers actually capture real power demand. We consider this point next.

3.3.2 Are datasheet power numbers accurate? As discussed in § 1, datasheet power numbers are primarily intended to dimension the power provisioning of a rack or server room. Thus, they are expected to be upper bounds of the actual router power draw in deployment. We aim to assess whether this expectation holds in practice and, if so, by how much datasheets overestimate the actual power draw.

Method. The SNMP dataset contains power traces from eight routers with "typical" or "maximum" power reported in their datasheet. We compute the median from each power trace and compare it to the datasheet number.

Result. Table 1 depicts the average measured power numbers, the datasheet value, and its relative overestimation compared to the measured value. As expected, for most models the datasheet overestimates the measured power. The surprise is that for the two models from the `Cisco 8000` series, the datasheet actually *underestimates* the measured power. There is nothing about the deployed routers for those two models that explains this; they run in cooled server rooms and are (very) far from operating at their maximum capacity. This provides an anecdotal example that datasheet numbers can be plainly wrong.

Summary. Datasheets do not obviously show energy efficiency improvements at the router level. Moreover, they provide only loose estimates that cannot be trusted as predictions of actual router power demands (**Q1**).

4 Router power model

The previous section confirmed that datasheet are poor predictors of actual router power demands. The other two data sources we study in this paper are direct power measurements and power models. Before assessing how accurate those are, we present how we define (§ 4.2) and derive (§ 5) our router power models.

4.1 Objectives

We aim to provide a modeling methodology that is easy to replicate on any router one has physical access to. Ideally, network operators could derive those models for their own routers. The resulting models should be able to accurately predict the effect of network management operations on power. Therefore, the power model should be (i) vendor agnostic; (ii) applicable to all router types; (iii) practical to derive, and thus should *not*

- rely on data that is not easily accessible, *e.g.*, the type of memory used to store routing tables;
- depend on dimensions that are hard to measure or vary in controlled ways, *e.g.*, temperature.

We propose a power model that focuses on *networking effects*, such as links being turned on or off, or traffic volume. While many other physical parameters influence a router's power draw, *e.g.*, temperature, aging, manufacturing quality, or the software version running on the device, we intentionally omit those parameters to keep the model practical to derive. We discuss the impact of that choice in § 4.3.

4.2 Model definition

We follow the approach from [35] and model router power as the sum of four main terms:

$P_{\textbf{base}}$ a base power term for turning the device on without any configuration;
$P_{\textbf{interface}}$ a cost for activating interfaces, depending on the transceiver module and the configured line rate;
$E_{\textbf{bit}}$ an energy cost per bit forwarded;
$E_{\textbf{pkt}}$ an energy cost per packet processed.

The first two terms capture the *static* part of the router power, P_{sta}, while the latter two capture the *dynamic* part, P_{dyn}, *i.e.*, the power that depends only on the traffic load. Both static and dynamic terms depend on the router configuration C, which is a vector specifying every interface configuration c_i. The dynamic term also depends on the load L, a vector specifying loads on every interface l_i. Therefore, we write

$$P = P_{\text{sta}}(C) + P_{\text{dyn}}(C, L) \tag{1}$$

$$P_{\text{sta}}(C) = P_{\text{base}} + \sum_{i=0}^{N_{port}} P_{\text{interface}}(c_i) \tag{2}$$

$$P_{\text{interface}}(c_i) = P_{\text{port}}(c_i) + P_{\text{trx}}(c_i), \tag{3}$$

where $P_{\text{port}}(c_i)$ denotes the power consumed by the router itself for an interface and $P_{\text{trx}}(c_i)$ is the power consumed by the transceiver, which is the pluggable module inserted into a port. Empirically, we observe that the transceiver power P_{trx} can be written as the sum of two terms: a fixed cost when the transceiver module is plugged into the router, $P_{\text{trx,in}}$, and a configuration-dependent cost once the interface is up, $P_{\text{trx,up}}$. We assume that the transceiver power cost is independent of the traffic load,[4] thus,

$$P_{\text{trx}}(c_i) = P_{\text{trx,in}} + P_{\text{trx,up}}(c_i) \tag{4}$$

Finally, the dynamic power is modeled as an affine function of the traffic load.

$$P_{\text{dyn}}(C, L) = \sum_{i=0}^{N_{port}} \left(P_{\text{port, dyn}}(c_i, l_i) + P_{\text{offset}}(c_i)\right) \tag{5}$$

$$P_{\text{port, dyn}}(c_i, l_i) = E_{\text{bit}}(c_i) \cdot r_i + E_{\text{pkt}}(c_i) \cdot p_i \tag{6}$$

where p_i and r_i are packet and physical layer bit rates, respectively, summed in both directions. Both terms are required to capture two independent energy-consuming actions: the processing of packet headers and the forwarding of bits to their output port. P_{offset} is a traffic-independent term we include to match empirical observations that the dynamic power is not strictly proportional to traffic. This can be due to some routers opportunistically turning off, *e.g.*, SerDes lines, when there is no traffic to forward. In other words, P_{offset} is the power difference between an interface carrying almost no traffic, *e.g.*, 1 pkt/s, and no traffic at all.

Each router can have ports of various types (*e.g.*, SPF+) which can host different transceiver types (*e.g.*, LR4). Each combination results in a different interface power profile. Thus, ultimately, this power model is composed of one constant term (P_{base}) and six terms per interface type and configuration: P_{port}, $P_{\text{trx,in}}$, $P_{\text{trx,up}}$, E_{bit}, E_{pkt}, and P_{offset}.

[4] Which makes sense for optical transceivers, where the laser remains on and dominates the power footprint. We validate that assumption in § 7.

4.3 Limitations

Generality. The router model we present in this paper focuses on fixed chassis routers; *i.e.*, it is not directly applicable to modular routers with pluggable linecards. In theory, it should be possible to extend the model by introducing a P_{linecard} term that could be measured similarly as P_{trx}. We leave this as future work.

Modeling scope. As introduced in § 4.1, the present model focuses on interfaces and traffic power. Several relevant power factors (see below) are not modeled and abstracted away into P_{base}. This is a purposeful choice that aims to make the model practical to derive by requiring only easy-to-collect data, such as traffic counters and transceiver models.

Omitted power factors. Our model does not account for several power-impacting factors, including

- environmental conditions (*e.g.*, temperature),
- fan speed (correlated to environmental conditions),
- power supply conversion losses,
- control plane processing,
- software version.

To include a factor in a model, one must be able to (i) control it to assess its power impact on the system, and (ii) measure that factor to derive the model predictions. The factors listed above are either hard to control, hard to measure, or both. Trying to account for such factors in the model would make it far more challenging to derive and, thus, less useful. Moreover, we postulate that the impact of such factors is either small (*e.g.*, the control plane processing) or pseudo-constant (*e.g.*, the temperature). If this is correct, then abstracting such factors within the P_{base} constant term should not hurt the model precision much but may result in some inaccuracy; *i.e.*, creating an offset to the true power value. A less intuitive but potentially important factor is the version of the operating system running on a router. Indeed, driver updates may, for instance, affect the efficiency of low-power modes of hardware components. During our deployment, we observed one such event where an operating system's update changed the temperature management logic, leading to an increased fan speed and a 45 W power bump, or approximately +12% (§ C, Fig. 8).

5 Modeling methodology

This section explains how one can derive the parameters of the power model presented in § 4.

The entire power modeling framework is open source [32].

5.1 Experimental setup

The experimental setup (Fig. 3) is composed of three main elements.

DUT The Device Under Test (DUT) is the router we aim to model.
Power meter A power meter that is controlled by and streams measurements over USB to the orchestrator. In this study, we use the Microchip MCP39F511N [25]. We chose this power meter as it offers a sufficient accuracy for our needs (specified at ±0.5%), is easy to use, features two measurement channels, and is compatible with standard C13 plugs while being reasonably small ($\approx 1/2dm^3$) and priced (≈\$280).

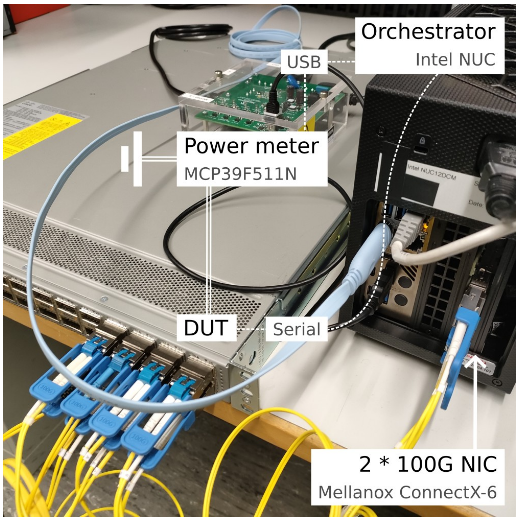


Figure 3: Our experimental setup for power modeling.

Orchestrator Finally, the orchestrator controls the power meter, configures the DUT over its console interface, and generates the test traffic required. In this study, we use an Intel NUC equipped with a Mellanox ConnectX-6 NIC. This solution enables generating 100 Gbps of unidirectional traffic while avoiding the high cost of a specialized traffic generator. We generate the traffic using the InfiniBand testing tool `ib_send_bw` for the larger bit rates (from 2.5 to 100 Gbps) and `iPerf3` in UDP mode for the smaller ones.

We assume that only one type of port (*e.g.*, QSFP) and transceivers (*e.g.*, LR4) are used for each experiment. This is not fundamental but simplifies the modeling methodology.

5.2 Derivation of model parameters

To derive the model parameters, one must perform five types of experiments:

Base The DUT is turned on, with no transceivers plugged in nor configuration.
Idle Transceivers are plugged in, but all ports are set down.
Port Ports are activated, but the interfaces remain down.
Trx Interfaces are up.
Snake Traffic is forwarded uniformly by all interfaces.

One can conveniently realize the **Idle**, **Port**, and **Trx** experiments with the same cabling, with DUT ports connected in pairs. For **Idle**, all ports are configured down. For **Port**, only one port in each pair is set up. For **Trx**, both ports are set up, which eventually takes the interfaces up. For **Snake** experiments, we configure the DUT to perform a layer 2 snake test, as defined in RFC 8239 [11]: the traffic is generated by the orchestrator, looped through all interfaces of the DUT and sent back to the orchestrator.

$P_{\textbf{<exp>}}$ is the total power measured during experiment **<exp>**, which can be expressed using the power model as follows. To simplify the writing, we assume that $2N$ interfaces are used, so N is the number of interface pairs.

$$P_{\textbf{Base}} = P_{\text{base}} \tag{7}$$
$$P_{\textbf{Idle}} = P_{\text{base}} + 2N \cdot P_{\text{trx,in}} \tag{8}$$
$$P_{\textbf{Port}} = P_{\textbf{Idle}} + N \cdot P_{\text{port}} \tag{9}$$
$$P_{\textbf{Trx}} = P_{\textbf{Idle}} + N \cdot (P_{\text{port}} + P_{\text{trx,up}}) \tag{10}$$
$$P_{\textbf{Snake}} = P_{\textbf{Trx}} + P_{\text{dyn}} \tag{11}$$

Eq. (7) gives P_{base} directly.

From Eq. (8), we derive $P_{\text{trx,in}} = (P_{\textbf{Idle}} - P_{\text{base}})/2N$.

From Eq. (9), we could derive P_{port} directly. However, to validate the linear behavior assumed by the model and avoid accumulating errors from the estimation of $P_{\textbf{Idle}}$, we instead measure $P_{\textbf{Port}}$ for multiple values of N and derive P_{port} via linear regression over N.

Similarly, from Eq. (10), we obtain $P_{\text{port}} + P_{\text{trx,up}}$ via linear regression over N on $P_{\textbf{Trx}}$ and deduce $P_{\text{trx,up}}$.

The derivation of E_{bit} and E_{pkt} is more involved and generally follows the approach proposed in [35]. The interface packet and bit rates are related by

$$p = r/(8 \cdot (L + L_{header})) \tag{12}$$

where L and L_{header} are the packet and header sizes in bytes, respectively. Thus, Eq. (6) rewrites:

$$P_{\text{port, dyn}} = E_{\text{bit}} \cdot r + E_{\text{pkt}} \cdot r/(8 \cdot (L + L_{header})) \tag{13}$$
$$\Rightarrow \frac{\partial P_{\text{port, dyn}}}{\partial r} = E_{\text{bit}} + E_{\text{pkt}}/(8 \cdot (L + L_{header})) \tag{14}$$

The model assumes that, for a fixed L, the power is a linear function of the bit rate r. In other words,

$$P_{\text{port, dyn}}|L = \alpha_L \cdot r + \beta_L \tag{15}$$
$$\Rightarrow \frac{\partial P_{\text{port, dyn}}|L}{\partial r} = \alpha_L \tag{16}$$

Since Eq. (14) is in fact the same as Eq. (16),

$$E_{\text{bit}} \cdot 8 \cdot (L + L_{header}) + E_{\text{pkt}} = \alpha_L \cdot 8 \cdot (L + L_{header}) \tag{17}$$

From Eq. (17), we can derive E_{bit} and E_{pkt} in two steps. First, we measure $P_{\textbf{Snake}}$ for a fixed L with varying bit rates r; α_L is the slope of the linear regression over r (Eq. (16)). We can repeat this process for multiple values of L, which allows computing multiple values for the right-hand side of Eq. (17). Second, we can plot those values and take another linear regression over L, which intercept and slope give E_{pkt} and $(E_{\text{bit}} \cdot 8 \cdot (L + L_{header}))$, respectively.[5]

Finally, we derive P_{offset} from the intercept β_L of the regression of $P_{\text{port,dyn}}$ over r:

$$P_{\text{offset}} = \beta_L - P_{\text{Trx}} \tag{18}$$

6 Validation of power numbers

We aim to assess if the in-router power measurements are trustworthy (**Q2**) and the power models derived in § 5 are accurate (**Q3**). To achieve this, we need a ground truth for the router power, which we obtain by measuring production routers externally.

[5]This derivation assumes a single interface i with configuration c_i. For better accuracy, we use as many equally-configured interfaces as possible, and then simply divide the coefficients α_L by the number of interfaces to get E_{bit} and E_{pkt} for a single interface.

Table 2: Example power models derived using the methodology described in § 5.
†: Due to the interface's low speed, the observable power variation with traffic is very small, rendering the derivation of E_{bit} and E_{pkt} imprecise. The resulting errors matter little for the same reason: traffic-induced power is small on this device.

	Port	Trans.	Speed	P_{base} [W]	P_{port} [W]	$P_{\text{trx,in}}$ [W]	$P_{\text{trx,up}}$ [W]	E_{bit} [pJ]	E_{pkt} [nJ]	P_{offset} [W]
(a)	`NCS-55A1-24H`									
	QSPF28	Passive DAC	100G	320	0.32	0.02	0.19	22	58	0.37
	|	|	50G	-	0.18	0.02	0.16	21	57	0.34
	|	|	25G	-	0.10	0.02	0.08	21	55	0.21
(b)	`Nexus9336-FX2`									
	QSFP28	LR	100G	285	1.9	2.79	-0.06	8	24	-0.43
	|	Passive DAC	100G	285	1.13	0.09	-0.02	8	26	0.07
(c)	`8201-32FH`									
	QSFP	Passive DAC	100G	253	0.94	0.35	0.21	3	13	-0.04
(d)	`N540X-8Z16G-SYS-A`									
	SFP	T	1G	33	-0.0	3.41	0.0	37	-48†	0.01

6.1 Autopower

Performing external power measurements of production routers pauses several operational constraints, including physical access to the routers. To facilitate the collection of such external power measurements, we designed a system called Autopower [13]. Ultimately, the objective is that network operators can autonomously deploy measurement units in their network—akin to RIPE Atlas probes [5] but for power measurements.

Requirements. An Autopower unit should be simple to deploy, ideally plug-and-play. One should be able to remotely control and retrieve measurement data. The system should be resilient to network or power interruptions and work even if the units are deployed behind a NAT. Finally, the power meter should be compatible with common router plugs.

Solution. A Autopower unit is composed of a Raspberry Pi 4 and a Microchip MCP39F511N [25] power meter, which features two C13 measurement channels. One channel is used to monitor a router's PSU while the other can be used to power the Pi; hence, Autopower units do not require additional power plugs, which are sometimes scarce in an ISP PoP. The power meter datasheet promises a measurement accuracy of ±0.5% which we experimentally validated using a high-end power meter (not shown). The Pi runs the client side of a custom client-server application to remotely control the power meter. The application is built using gRPC [3] and a client-initiated connection to the server in order to work behind a NAT. Thus, the client only requires an outgoing connection to the Internet.[6] The client locally stores the power measurements with periodic uploads to the server. The connection to the server and the power measurement start automatically on boot, which provides resilience against power failures—which did occur once over our deployment period.

[6]More specifically: to our server's IP on a custom port.

The Autopower software is open-source [13] and includes additional features such as Zabbix monitoring for the clients and a web interface to conveniently start/stop measurements or download the power data (see Fig. 7, § C). Refer to [12] for more details about Autopower.

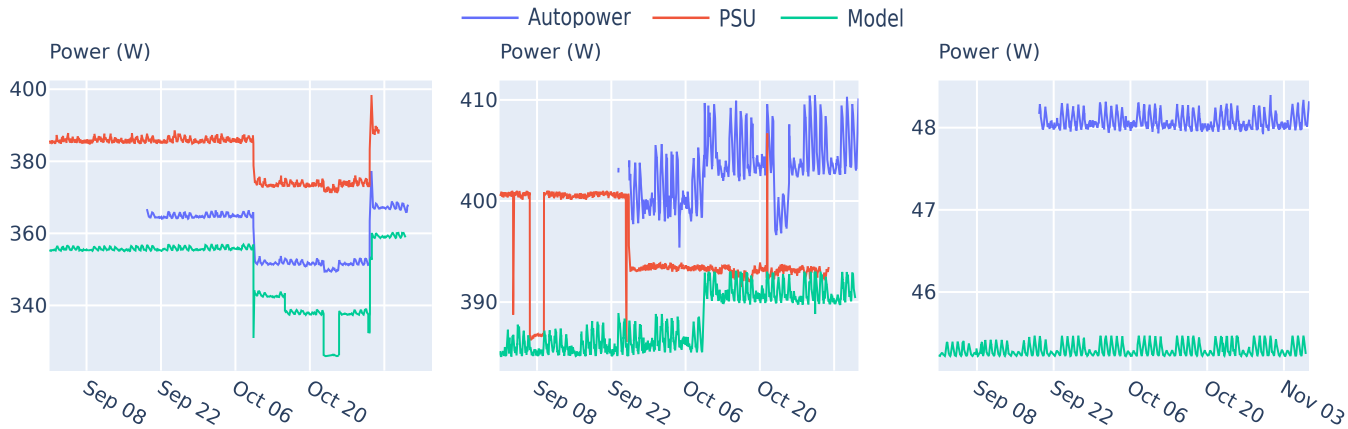
6.2 Comparing the power data sources

Method. We collected Autopower data over two months for three different models of routers in deployment. We also performed all the lab measurements required to derive power models for those routers. We combine those models with the deployed routers' module inventory files (giving the transceiver module types) and the traffic counters (packet and bytes) to derive model predictions of the power draw over time. This allows, for the same physical device, a direct comparison of (i) PSU measurements, (ii) external measurements, and (iii) power model predictions. To smooth out the noise in the power measurements, we plot 30-minute averaged traces in Fig. 4. See § C, Fig. 9 for zoomed versions.

Result – PSU vs Autopower. The PSU measurements do not consistently match the external ones, which we take as ground truth. In Fig. 4a, the shape matches well but has a 15–20 W offset. In Fig. 4b, the shape does not match. It appears more as a pseudo-constant value with sharp jumps. Fig. 4c does not show PSU measurements because this router model does not report its power draw.

Result – Model vs Autopower. The shapes of the power models consistently match the external measurements: traffic-induced power fluctuations perfectly match in time (for all three) and magnitude (Fig. 4a, 4c). However, we note a consistent underestimation of ≈ 9 W, 13 W and 3 W, respectively.

Discussion. Digging into the details allows explaining some of the events visible in the traces. In Fig. 4a, on Oct. 9, a 400G interface is removed, which previously contained a 400G FR4 transceiver; all traces drop by ≈ 13 W, which matches with the 12 W specified for that transceiver plus ≈ 1 W of P_{port} predicted by the model (Table 2,



(a) `8201-32FH` **(b) `NCS-55A1-24H`** **(c) `N540X-8Z16G-SYS-A`**

Figure 4: Comparison of PSU measurements, Autopower measurements, and the predictions made by our power models. ***Fig. 4c does not show PSU measurements because this router model does not report its power draw.***

(c)). Similarly, multiple interfaces are added on Oct. 31. Between Oct. 22 and 25, all traces dropped, but the model more than the measurements. A closer inspection of the event logs reveals that one interface experienced regular flapping; on the 22nd, the interface was manually taken down to fix the issue and then put back up on the 25th. The difference in magnitude in power drops is because the model assumes the transceiver was unplugged, though—apparently—it was not. We will discuss this phenomenon further in the next section. The Switch operators confirmed that transceivers are sometimes left plugged into inactive ports to be used either as spares or awaiting pick-up at the next PoP visit; those spare transceivers may partly explain the offset of the model.

In Fig. 4b, the PSU measurements show one 7 W drop on Sept. 25, which coincides with the deployment of our Autopower units. To install a power meter, we need to shortly unplug each power supply before re-plugging it into the meter. Closer inspection of the raw data shows that one of the two PSUs suddenly reported 7 W less than before while *nothing* had changed besides the power cycling. Combined with the prior observation that the PSU measurements of that router do not capture the traffic-induced patterns, the only conclusion we can draw is that those PSU measurements simply cannot be trusted.

Summary. For the routers we studied, we validate for the first time on real-world data that analytical power models can precisely estimate router power demand, though with an offset (**Q3**). Moreover, routers' PSU measurements cannot be universally trusted (**Q2**). For the two devices we could test, one is precise (*i.e.*, captures the shape) but not accurate (*i.e.*, is offset), and the other is neither. Replications of this study are necessary to assess the generality of those observations.

7 Insights on router power

Now that we have confirmed the validity of power model predictions, we can analyze those models to reveal several interesting insights on router power.

"Down" does not mean "Off." This is possibly the most important insight we gain from the power model: while it seems intuitive that turning down an interface would power off any transceiver plugged in, this is not the case—at least not in the routers we studied. A similar observation was made in [28]. The power model captures this with the term $P_{\text{trx\_in}}$, which is the observed increase in power when a transceiver is plugged into a deactivated port. For the optical transceivers in our experiments, $P_{\text{trx\_in}}$ dominates the total transceiver power ($P_{\text{trx\_in}} + P_{\text{trx\_up}}$) (Table 2). It is not clear if there is a technical reason for this behavior. In a series of blog posts [6–8], Juniper's engineers explain that the default behavior on some of their routers is to keep the SerDes lines up even when ports are down (!) and suggest using a specific configuration to power them off. Their experiments show this can result in a whooping 47% reduction in the base power of a `ACX7100-32C` router. We postulate that, similarly, keeping transceivers powered up is an arbitrary choice of the operating system—and thus could be easily fixable.

Models need correct inputs. As discussed in § 6, Fig. 4a illustrates a case where the model wrongly assumes that an interface is disabled while, in fact, a power-hungry transceiver is still plugged in and drawing power. Obviously, without correct or complete input data, models cannot magically predict correct power values. More fundamentally, this illustrates that there may be more inputs necessary for an accurate prediction than anticipated. In § 6.2, we use the presence of traffic counters for a given interface as signaling that the interface is active. This reasoning is correct, but the negative is not: an interface might be drawing power despite reporting no traffic counters.

The energy cost of traffic is small. The energy cost per bit, E_{bit}, and per packet, E_{pkt}, are relatively consistent on high-speed ports, in the range of a few pJ per bit and nJ per packet, respectively. Converted into macroscopic units, and assuming average values of 5 pJ per bit and 15 nJ per packet, forwarding 100 Gbps of traffic demands between 3.4 and 0.6 W for 64 B and 1500 B packets, respectively. At that rate, forwarding the total Switch traffic would cost only 5.9 W; *i.e.*, 0.02% of the total network power.[7]

The energy cost of transceivers is significant. In contrast, all the transceivers in the Switch network collectively draw 2.2 kW; that is, ≈10% of the total network power.

Transceiver power is independent of the traffic load. In § 4, we make the assumption that the transceiver power is independent of the traffic load. This assumption justifies using the datasheet transceiver power value to estimate P_{trx} in the power model validation (§ 6.2). The power models we derived are compatible with that assumption. Specifically, on Table 2 (b), we observe that the value for E_{bit} is approximately equal whether we use an optical or a passive electrical transceiver. This means that either the transceiver power is indeed constant, or the power variation with traffic is the same for both transceivers.[8] Otherwise, we would see different numerical E_{bit} values for different transceivers.

8 Power savings of link sleeping

Saving power by turning off links that are not needed to carry the total traffic was proposed in several prior works [27, 31, 37]. To their own admission, those works estimated power savings using high-level power models with little connection to real-world router power demand. They did so because more fine-grained or realistic power models were not available. With this gap now filled, we can derive better estimates for the power savings of link sleeping.

Method. First, we use the code provided by the authors [30] to apply the Hypnos algorithm [31] to our dataset. Then, we combine the output with (i) our power model predictions and (ii) inventory files listing transceiver modules per interface.

Intuitively, one would expect to save Eq. (3): $P_{\mathrm{interface}} = P_{\mathrm{port}} + P_{\mathrm{trx}}$ on each side of a link that is turned off. However, as discussed in § 7, a share of the transceiver power, $P_{\mathrm{trx\_in}}$, is paid as soon as the transceiver is plugged into a port, even if that port is off. Thus, our model predicts turning off a port saves $P_{\mathrm{port}} + P_{\mathrm{trx\_up}}$. Since we do not have power models for every router and transceiver deployed in the Switch network, we must make further simplifications. First, we assume a constant value of P_{port} per port type (§ B, Table 5). We get those values by averaging all the power models we have per port type.[9] Second, we use transceiver datasheet values to estimate P_{trx}. Without transceiver power models, we do not know how P_{trx} is split between $P_{\mathrm{trx\_in}}$ and $P_{\mathrm{trx\_up}}$. We can only say that $P_{\mathrm{trx\_up}} \in [0, P_{\mathrm{trx}}]$, which results in a range of expected power savings.

Result. We run Hypnos on the Switch network traces over a span of one month and find that the algorithm would save between 80 and 390 W, or 0.4–1.9% of the total router power.

Discussion. In their evaluation, the Hypnos' authors find the algorithm can turn off around one-third of the links and postulate that this would save about one-third of the total transceiver power, plus some additional router power—what we call P_{port} in § 4.

Our power model suggests that things are more complex. First, our modeling experiments clearly show that a share of the transceiver power is paid as soon as the transceiver is plugged in ($P_{\mathrm{trx\_in}}$); that share dominates for the optical transceivers we tested and is unaffected by turning the interface down. Thus, while we do not have all the required model data to assert this, we postulate that the actual power savings will be closer to the lower end of our estimation. Moreover, Hypnos (and similar prior proposals) are intra-domain protocols; they can only turn off *internal* links. In the Switch data we study, 51% of the interfaces are external, *i.e.*, connecting to other networks, and represent 52% of the total transceiver power. Link sleeping, as considered thus far in the literature, is only applicable to internal links, which may be the minority in Tier-2 or -3 ISPs.

Summary. The savings from turning links off are limited by (i) the ratio of internal links in the network's total power and (ii) whether routers physically power off transceivers when taking an interface down.

9 PSU analysis

One observation from Fig. 4 was that power models precisely estimate the actual power, but with an offset. As discussed in § 4.3, such an offset is not surprising given the power factors that are not explicitly modeled. We postulate that different power conversion efficiencies of the routers' PSUs may explain this offset in part. We present preliminary data to test that hypothesis and, observing poor efficiencies for certain routers, we further explore the potential of PSU optimization as a power-saving approach at the router level.

9.1 Background on PSU efficiency

Every electrical device requires electricity supplied at the correct voltage. A power supply unit (PSU) is the component responsible for converting the electricity from the power outlet form (*e.g.*, 230V AC) to the voltage required by the device; for routers, that is typically 12V DC. It is common for routers to have two (or more) PSUs for redundancy reasons.

This power conversion is not perfect; it comes with power losses that depend on the efficiency of the PSU. The efficiency of a PSU is measured as the ratio between the power delivered by the PSU, P_{out}, divided by the power feeding into the PSU, P_{in}, and is generally expressed as a percentage. Moreover, the PSU efficiency is not a constant; it varies with the load and is usually best at around 50–60% of the PSU capacity and notoriously bad at loads below 10–20%. Fig. 5 shows one PSU efficiency curve as an example. The "80 Plus" standard [9] was introduced in 2004 to foster PSU efficiency improvements. The name comes from the initial requirement of

[7]The real traffic cost is a bit higher (though still tiny) because a lot of the Switch traffic goes over 10G ports, which are less energy efficient.
[8]We could test this on a couple of routers only. More experiments would be useful to confirm this observation.
[9]Our measurements show P_{port} varies across router models (Table 2)

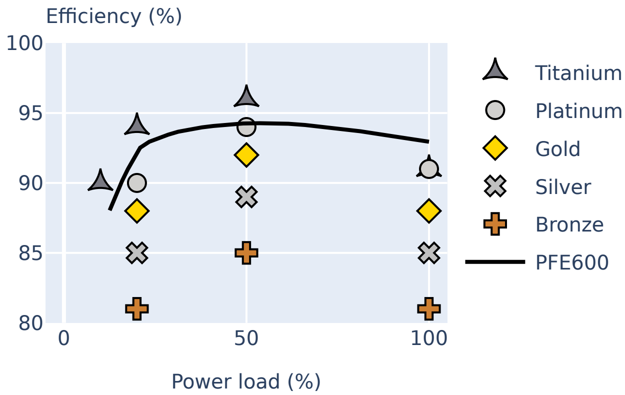


Figure 5: The efficiency curve of the *Platinum*-rated `PFE600-12-054xA`, the PSU found in the `Wedge 100BF-32X`, redrawn from the PSU datasheet. The 80 Plus standard set points are also shown. To be certified with an 80 Plus standard, a PSU must have an efficiency larger than the corresponding set points.

achieving an efficiency larger than 80% at loads of 20%, 50%, and 100%. The standard has since been extended with higher requirement levels ranging from *Bronze* to *Titanium*, illustrated in Fig. 5.

9.2 Dataset

To study the impact of PSU efficiency on router power, we combine the SNMP PSU measurement traces from our dataset with

- a list of the PSU capacities in the Switch routers (*i.e.*, the maximum power they can deliver), obtained via the hardware inventory reported by the routers;
- a one-time export of P_{in} and P_{out} of the routers' PSUs, obtained via a snapshot of the environment sensor readings reported by the routers (our SNMP traces only contain P_{in}).

We cannot share the source inventory and sensor data, but make all the data used in the following analysis available [20].

We combined the P_{in} and P_{out} readings with the PSU capacities to compute the efficiency and load of each PSU in our dataset. Note that this provides only *one* snapshot of the PSU efficiency. Moreover, it is unknown how precise the sensor readings are and how frequently they are updated. Some PSUs reported values of P_{out} larger than P_{in}, which is physically impossible; it may be due to poor sensor quality and/or asynchronous measurements of the two power values. In those cases, we cap the efficiency at 100%.

9.3 Investigating PSUs (in)efficiency

9.3.1 How efficient are the router PSUs today? Fig. 6 shows the efficiency of each PSU in our dataset, grouped by router models. We make several important observations.

First, the PSU loads are low, between 10 and 20%. This is not surprising given that (i) the traffic load on those routers is low (Fig. 1), thus the power draw is expected to be far from its maximum, and (ii) two PSUs are used for redundancy.

Second, the efficiency numbers vary widely, from very good (> 95%) to very poor (< 70%). As expected, there is variability across router models: *e.g.*, the `NCS-55A1-24H` fares rather well, with efficiencies generally above 85% (Fig. 6b); conversely the efficiency for the `8201-32FH` PSUs is only 76% or worse (Fig. 6c). Perhaps more surprisingly, there is also a large variability across routers of the same model. This is most obvious for the `ASR-920-24SZ-M` for which the efficiency numbers span the entire range of our dataset (Fig. 6d). One possible reason for that variability could be the operating temperature, but, in our dataset, there is no apparent correlation between temperature and PSU efficiency (not shown). Since all PSUs are of the same model, other possible explanations include aging effects or differences in manufacturing quality. In any case, the data suggests that improving PSUs' efficiency or, more generally, reducing the power conversion losses is a promising power saving measure. We now aim to estimate the potential gains for the Switch network.

9.3.2 How much would we save with more efficient PSUs? We first estimate the power savings if every PSU in the Switch network were at least as efficient as the five 80 Plus standards.

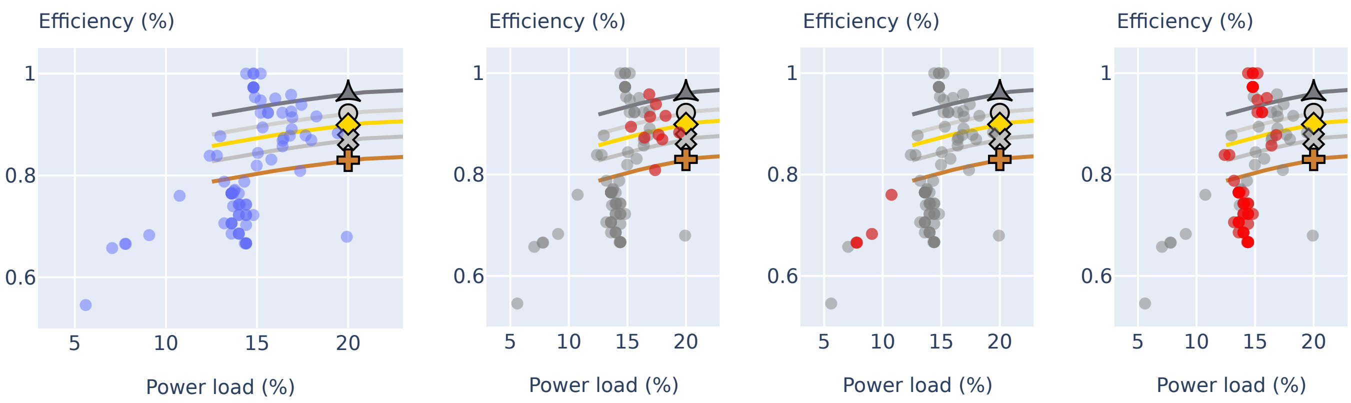
Method. To estimate the power gains from more efficient PSUs, we need a model of the efficiency curve. We assume that the efficiency curve of any PSU is the same as the `PFE600` curve (Fig. 5) plus a constant offset. Thus, we can derive a theoretical efficiency curve for each standard. For each PSU in the Switch dataset, we have one efficiency point and the corresponding power load. Thus, we can compute the power savings if the efficiency of each PSU were to "rise" to each of the 80 Plus standards.

Results. On average, over the entire network, we find that having at least *Bronze*-rated PSUs would yield 2% power savings, 5% for *Platinum* (today's best commonly available PSUs), and up to 7% for *Titanium* (Table 3).

9.3.3 How much would we save with better-sized PSUs? The data shows that most PSUs operate under low load, where they are known to be least effective. As discussed in § 1, this is in part because the power supply gets provisioned from datasheet power numbers, which tend to overestimate deployment power demands. Here, we try to estimate the power cost of this overprovisioning.

Method. We consider the five PSU capacities present in the dataset, ranging from 250W to 2700W. For each router, we consider the maximum load over the PSUs, l_{max}, and define the minimum PSU capacity C among the five available such that $C \geq k * l_{max}$. Setting $k = 2$ implies maintaining resilience to one PSU failure. $k = 1$ will increase savings but reduce the power provisioning margins. We then estimate the power impact of resetting every power supply capacity to the maximum of C and each of our five capacity options. Small sizes should yield power savings with each PSU operating at a better efficiency point, while larger ones would induce smaller PSU loads and thus increase inefficiency.

Results. The results are summarized in Table 4. As expected, the smaller k and the PSU capacity considered, the larger the savings. Conversely, the power starts to increase when we set all PSU capacities to at least 1100W. However, it is worth noting that the differences are small, *i.e.*, ranging from 2% in the "best" case ($k = 1$, 250W capacity) to -1% in the "worst" case ($k = 2$, 2700W capacity). This indicates that the cost of over-dimensioning the PSUs is



(a) All PSU efficiency points (b) `NCS-55A1-24H` (c) `8201-32FH` (d) `ASR-920-24SZ-M`

Figure 6: The PSU efficiency values span a large spectrum from very good (> 95%) to very poor (< 60%). Some router models fare particularly well (Fig. 6b), some particularly bad (Fig. 6c), and some vary (Fig. 6d).

Table 3: Using more efficient power supply and using only one are promising vectors of energy savings.

Power-saving measure ╲ 80 Plus standard	Bronze	Silver	Gold	Platinum	Titanium
§ 9.3.2: More efficient PSUs	2% (482 W)	3% (737 W)	4% (958 W)	5% (1156 W)	7% (1563 W)
§ 9.3.4: Only one PSU	4% (1002 W)	–	–	–	–
§ 9.3.5: Both	5% (1240 W)	6% (1392 W)	7% (1528 W)	7% (1660 W)	9% (1974 W)

Table 4: It is best to size the PSU capacity as closely to the required power (leftmost columns & $k = 1$). However, the power cost of over-dimensioning is small (rightmost columns & $k = 2$).

k ╲ Minimum PSU capacity	250 W	400 W	750 W	1100 W	2000 W	2700 W
$k = 1$	2% (520 W)	2% (456 W)	1% (287 W)	0% (-21 W)	-1% (-247 W)	-1% (-247 W)
$k = 2$	2% (502 W)	2% (432 W)	1% (287 W)	0% (-21 W)	-1% (-247 W)	-1% (-247 W)

limited; more specifically, over-dimensioning costs less than poor efficiency (§ 9.3.2).

9.3.4 How much would we save by loading only one PSU. Another approach to optimize PSUs efficiency is to stop load-balancing power among PSUs and use only one.[10]

Method. As before, we assume that the efficiency curve of any PSU is the same as the `PFE600` curve (Fig. 5) plus a constant offset. We compute that constant from the efficiency data point for each PSU, then obtain the new efficiency value of that PSU if it would deliver the total router power, thus operating at (roughly) twice its load. We assume zero power losses from the second PSU.

Results. On average, over the entire network, using one PSU instead of two is estimated to save 4% of the total power.

9.3.5 How much would we save with more efficient PSUs AND loading only one? We can push things further and estimate the effect of combining the two most effective measures we have considered so far: using only one PSU *and* assume that PSU meets high(er) efficiency standards.

Method. We combined the methods of § 9.3.2 and § 9.3.4.

Results. As expected, the savings of both measures roughly add up, resulting in gains of 5% for using only one *Bronze*-rated PSUs up to 9% for *Titanium* (Table 3).

9.4 Discussion

Based on our analysis of the PSU data of the Switch network, we formulate the following takeaway points.

- We cannot assume that the efficiency of router PSUs is good; empirical evidence shows it can be poor.
- Improving the average efficiency of PSUs can yield sizable savings (2–7%).
- Operating PSUs at medium load optimizes their efficiency; the PSU capacity should be selected to match the expected power demand. However, over-dimensioned PSUs are less costly than PSUs with poor efficiency.

[10]The second PSU may remain online to maintain the supply redundancy; this is a standard technique known as "hot stand-by." Unfortunately, none of the routers we studied supported that feature.

- If PSUs are over-dimensioned (which was the case throughout our dataset) using only one PSU instead of two can yield sizable savings (4%). According to private correspondence with power electronic researchers, there does not seem to be any technical limitation to implementing "hot stand-by" capabilities in today's routers, which would enable those savings while maintaining resiliency to PSU failures.
- Network monitoring tools should include both input and output PSU power to enable PSU efficiency tracking over time. This was not the case in our dataset, which is why we relied on only a snapshot of the router's sensor readings.

Ultimately, reducing power conversion losses in PSUs appears to be a promising energy-saving vector. However, further research is necessary. First, we could only coarsely estimate the shape of the efficiency curves for the PSUs in our dataset, based on one efficiency profile and one efficiency data point. A better understanding of how those curves look in practice is necessary to improve the estimates. Second, we must consider the potential side effects of enabling some "hot stand-by" mode on routers; *e.g.*, it may be that thermal differences between PSUs lead to increased hardware wear and tear. There is currently no evidence either way that we know of. History shows that thermal effects are complex and need to be measured rather than analytically predicted [15].

10 Discussion and future work

This work aims to assess which data can be used to realistically predict router power demands. We confirmed the intuition that datasheets do not provide reliable estimates (§ 3). We also observed that internal router measurements cannot always be trusted; it appears some routers provide usable power values, while others do not (§ 6). We must study a larger array of routers to assess the generality of this problem. Finally, we derived empirical power models similar to prior proposals from the literature [35] and validated on two different routers that they provide precise predictions, albeit with a constant offset to the true router power (§ 6).

In addition, our analysis of the data we collected suggests some opportunities for power savings. First, the routers we studied do not power off transceivers when taking down ports (§ 7), which was independently observed as well in [28]. This should be addressable in software, though confirmation demands further research.

In Switch, the ISP network we study, the total transceiver power accounts for ≈10% of the total power (§ 7). The low utilization of the links in that network (Fig. 1) makes it ripe for "link sleeping," *i.e.*, turning links off to save power, an old idea [27, 37] with a recent resurgence in interest [28, 31]. However, our power model suggests that this likely yields lower savings than anticipated (§ 8); in part because of the previous point (transceivers not being actually powered off), but also because, in the Switch network, only half of the transceivers are connected internally and thus feasible to turn off. It is not yet clear how common that situation is.

Finally, we start to explore the potential of using more efficient and better-dimensioned power supply units to minimize power conversion losses, which appears promising (§ 9). The available data to study PSU efficiency is limited today. It requires measurements of both the power flowing in and delivered by a PSU. The latter can only be performed by the PSU and, as discussed above, it is unclear whether those measurements can be trusted. Moreover, standard network monitoring protocols currently do not support the collection of both power values; this issue will hopefully be addressed by the newly-formed IETF GREEN working group [2].

This paper presents a diverse set of router power data, but only scratches the surface. Practical constraints limited our study to a handful of router models, mostly from the same vendor. It is imperative that similar data be collected from a larger set of routers and deployment contexts.

> We aimed to facilitate future data collection efforts by releasing our tools [13, 32, 36] and building Network Power Zoo [18] to serve as a central repository of router power data.

11 Ethics

All the data from Switch was collected after approval and signature of a data sharing agreement. It is being published with explicit agreement from Switch. Given the large user population of Switch and the respective 5min and 0.5s resolution of the SNMP and Autopower measurements, individual users' privacy is preserved. Prior to publication, the router names have been anonymized to obfuscate the physical locations (encoded into the original names) but preserve the relations between routers; *i.e.*, anonymized names identify routers deployed in the same point of presence.

Acknowledgments

We thank the entire Switch NOC team for their support and approval of the data release. This work was supported by the ETH Future Computing Laboratory (EFCL), financed by a donation from Huawei Technologies.

References

[1] [n. d.]. Cisco 8000 Series Routers Data Sheet. https://www.cisco.com/c/en/us/products/collateral/routers/8000-series-routers/datasheet-c78-742571.html

[2] [n. d.]. Getting Ready for Energy-Efficient Networking (Green). https://datatracker.ietf.org/wg/green/about/

[3] [n. d.]. gRPC. https://grpc.io/

[4] [n. d.]. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/

[5] [n. d.]. RIPE Atlas - RIPE Network Coordination Centre. https://atlas.ripe.net/

[6] 2023. PTX10000 Power Optimization. https://community.juniper.net/blogs/nicolas-fevrier/2023/08/29/ptx10000-power-optimization

[7] 2023. Saving Energy on PTX with PFE Power Off. https://community.juniper.net/blogs/ramdas-machat/2023/07/27/saving-energy-on-ptx-with-pfe-power-off

[8] 2023. Saving Power on ACX7000 Series. https://community.juniper.net/blogs/nicolas-fevrier/2023/09/11/saving-power-on-acx7000-series

[9] 2024. 80 Plus. *Wikipedia* (Sept. 2024). https://en.wikipedia.org/w/index.php?title=80_Plus&oldid=1244462480

[10] 2024. Device Type Library. NetBox Community. https://github.com/netbox-community/devicetype-library

[11] Lucien Avramov and jhrapp@gmail.com. 2017. *Data Center Benchmarking Methodology.* Request for Comments RFC 8239. Internet Engineering Task Force. https://doi.org/10.17487/RFC8239

[12] Jonathan Chung. 2024. *Automated Power Measurement for Network Devices: Collecting Data Reliably Made Simple?* Bachelor Thesis. ETH Zurich. https://doi.org/10.3929/ethz-b-000685212

[13] Jonathan Chung. 2025. Autopower. Networked Systems Group (NSG). https://github.com/nsg-ethz/autopower

[14] David de la Osa Mostazo, Pablo Armingol Robles, Óscar González de Dios, and Juan Pedro Fernández-Palacios Giménez. 2024. Lessons Learned from IP Routers Power Measurements and Characterization. In *2024 15th International Conference on Network of the Future (NoF).* 245–253. https://doi.org/10.1109/NoF62948.2024.10741444

[15] Nosayba El-Sayed, Ioan A. Stefanovici, George Amvrosiadis, Andy A. Hwang, and Bianca Schroeder. 2012. Temperature Management in Data Centers: Why Some (Might) like It Hot. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE*

Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '12). Association for Computing Machinery, New York, NY, USA, 163–174. https://doi.org/10.1145/2254756.2254778

[16] S. El-Zahr, P. Gunning, and N. Zilberman. 2023. Exploring the Benefits of Carbon-Aware Routing. *Proceedings of the ACM on Networking (PACMNET)* (2023). https://ora.ox.ac.uk/objects/uuid:504d0a87-6742-414f-b2c2-be74f2e5b579

[17] Maruti Gupta and Suresh Singh. 2003. Greening of the Internet. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*. Association for Computing Machinery, New York, NY, USA, 19–26. https://doi.org/10.1145/863955.863959

[18] Andreas Hunziker and Romain Jacob. 2024. Network Power Zoo. https://networkpowerzoo.ethz.ch/

[19] IEA. [n. d.]. Data Centres & Networks. https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

[20] Romain Jacob. 2025. Fantastic Joules and Where to Find Them: Software Artifacts. https://github.com/nsg-ethz/Fantastic-Joules-and-Where-to-Find-Them

[21] Itzik Kiselevsky. 2023. Evolution of Switches Power Consumption. https://eng.ox.ac.uk/media/11vdkdtb/itzikk_evolution-of-switches-power-consumption.pdf

[22] Youcef Magnouche, Jérémie Leguay, and Feng Zeng. 2023. Safe Routing in Energy-aware IP Networks. In *2023 19th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, Vilanova i la Geltru, Spain, 1–8. https://doi.org/10.1109/DRCN57075.2023.10108152

[23] Priya Mahadevan, Sujata Banerjee, and Puneet Sharma. 2010. Energy Proportionality of an Enterprise Network. In *Proceedings of the First ACM SIGCOMM Workshop on Green Networking (Green Networking '10)*. Association for Computing Machinery, New York, NY, USA, 53–60. https://doi.org/10.1145/1851290.1851302

[24] Jens Malmodin, Nina Lövehagen, Pernilla Bergmark, and Dag Lundén. 2024. ICT Sector Electricity Consumption and Greenhouse Gas Emissions – 2020 Outcome. *Telecommunications Policy* (Jan. 2024), 102701. https://doi.org/10.1016/j.telpol.2023.102701

[25] Microchip. [n. d.]. MCP39F511N. https://www.microchip.com/en-us/development-tool/ADM00706

[26] David Mytton, Dag Lundén, and Jens Malmodin. 2024. Network Energy Use Not Directly Proportional to Data Volume: The Power Model Approach for More Reliable Network Energy Consumption Calculations. *Journal of Industrial Ecology* n/a, n/a (2024). https://doi.org/10.1111/jiec.13512

[27] Sergiu Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall. 2008. Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In *5th USENIX Symposium on Networked Systems Design and Implementation (NSDI 08)*. https://www.usenix.org/conference/nsdi-08/reducing-network-energy-consumption-sleeping-and-rate-adaptation

[28] Daniel Otten, Alexander Brundiers, Timmy Schüller, and Nils Aschenbruck. 2023. Green Segment Routing for Improved Sustainability of Backbone Networks. In *2023 IEEE 48th Conference on Local Computer Networks (LCN)*. IEEE Computer Society, 1–9. https://doi.org/10.1109/LCN58197.2023.10223317

[29] Daniel Otten, Sebastian Neuner, and Nils Aschenbruck. 2023. On Modelling the Power Consumption of a Backbone Network. In *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. 1842–1847. https://doi.org/10.1109/ICCWorkshops57953.2023.10283615

[30] Lukas Röllin. 2024. Hypnos. Networked Systems Group (NSG). https://github.com/nsg-ethz/hypnos

[31] Lukas Röllin, Romain Jacob, and Laurent Vanbever. 2024. A Sleep Study for ISP Networks: Evaluating Link Sleeping on Real World Data. In *HotCarbon*. https://doi.org/10.3929/ethz-b-000680859

[32] Priska Steinebrunner, Jackie Lim, and Romain Jacob. 2025. NetPowerBench. Networked Systems Group (NSG). https://github.com/nsg-ethz/NetPowerBench

[33] Seyedali Tabaeiaghdaei, Simon Scherrer, Jonghoon Kwon, and Adrian Perrig. 2023. Carbon-Aware Global Routing in Path-Aware Networks. In *Proceedings of the 14th ACM International Conference on Future Energy Systems (E-Energy '23)*. Association for Computing Machinery, New York, NY, USA, 144–158. https://doi.org/10.1145/3575813.3595192

[34] Nedeljko Vasić and Dejan Kostić. 2010. Energy-Aware Traffic Engineering. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (E-Energy '10)*. Association for Computing Machinery, New York, NY, USA, 169–178. https://doi.org/10.1145/1791314.1791341

[35] Arun Vishwanath, Kerry Hinton, Robert W. A. Ayre, and Rodney S. Tucker. 2014. Modeling Energy Consumption in High-Capacity Routers and Switches. *IEEE Journal on Selected Areas in Communications* 32, 8 (Aug. 2014), 1524–1532. https://doi.org/10.1109/JSAC.2014.2335312

[36] Weiran Wang. 2024. Router Datasheet Parser. Networked Systems Group (NSG). https://github.com/nsg-ethz/router_datasheet_parser

[37] Mingui Zhang, Cheng Yi, Bin Liu, and Beichuan Zhang. 2010. GreenTE: Power-aware Traffic Engineering. In *The 18th IEEE International Conference on Network Protocols*. 21–30. https://doi.org/10.1109/ICNP.2010.5762751

A CRediT Statement

This section lists the author's contributions to this work. The contributions are described using CRediT, the Contributor Roles Taxonomy, an ANSI/NISO standard.

All authors agree with this declaration of contributions.

Romain Jacob	0000-0002-2218-5750
Conceptualization	Lead
Data curation	Lead
Formal analysis	Lead
Funding acquisition	Supporting
Investigation	Lead
Methodology	Lead
Project administration	Lead
Software	Lead
Supervision	Equal
Validation	Lead
Visualization	Lead
Writing – original draft	Lead
Writing – review & editing	Lead

Lukas Röllin	0009-0005-0720-5134
Formal analysis	Supporting
Investigation	Supporting
Software	Supporting
Validation	Supporting
Writing – original draft	Supporting
Writing – review & editing	Supporting

Jackie Lim	0009-0002-6011-8521
Investigation	Supporting
Methodology	Supporting
Software	Supporting

Jonathan Chung	0009-0004-4351-1879
Data curation	Supporting
Investigation	Supporting
Software	Supporting

Maurice Béhanzin	0009-0002-9620-7606
Formal analysis	Supporting
Investigation	Supporting
Software	Supporting
Visualization	Supporting

Weiran Wang	0009-0008-8315-8204
Data curation	Supporting
Investigation	Supporting
Software	Supporting

Andreas Hunziker	0009-0006-7227-0802
Data curation	Supporting
Software	Supporting

Theodor Moroianu	0009-0001-5041-2201
Investigation	Supporting
Software	Supporting

Seyedali Tabaeiaghdaei	0000-0002-7629-1874
Investigation	Supporting
Methodology	Supporting
Software	Supporting
Supervision	Supporting
Writing – original draft	Supporting
Writing – review & editing	Supporting

Adrian Perrig	0000-0002-5280-5412
Conceptualization	Supporting
Funding acquisition	Equal
Resources	Equal
Supervision	Equal
Writing – original draft	Supporting
Writing – review & editing	Supporting

Laurent Vanbever	0000-0003-1455-4381
Conceptualization	Supporting
Funding acquisition	Equal
Resources	Equal
Supervision	Equal
Writing – original draft	Supporting
Writing – review & editing	Supporting

B Additional power models

- Table 6 summarizes additional power models we derived using the methodology described in § 5 but not explicitly discuss in this paper.
- Table 5 lists the P_{port} and P_{trx_in} used per port type in the link sleeping evaluation (§ 8).

C Additional visuals

- Fig. 9 shows a zoomed-in version of Fig. 4 where the power model is manually offset to the level of the Autopower measurements to show the precision of the model predictions.
- Fig. 7 shows the web user interface to conveniently control Autopower measurement units.
- Fig. 8 shows the PSU measurements reported by a Cisco `8201-32FH` router across an OS update.

Table 5: P_{port} and P_{trx_in} used per port type in the link sleeping evaluation (§ 8).

Port type	P_{port} (W)	P_{trx_up} (W)
SFP	0.05	0.005
SFP+	0.55	-0.016
QSFP28	0.53	0.126
QSFP-DD	1.82	-0.069

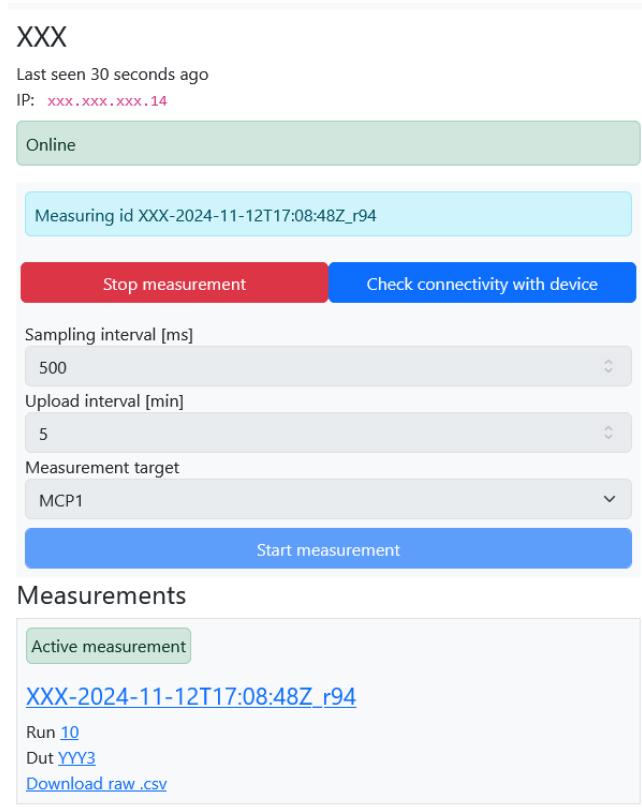


Figure 7: The web interface allows remote control of a Autopower client and downloading measurement data.

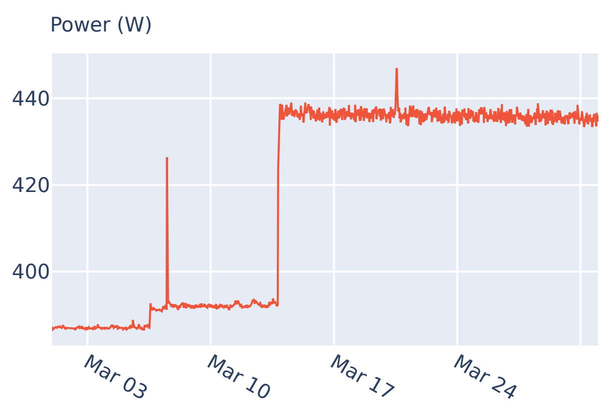
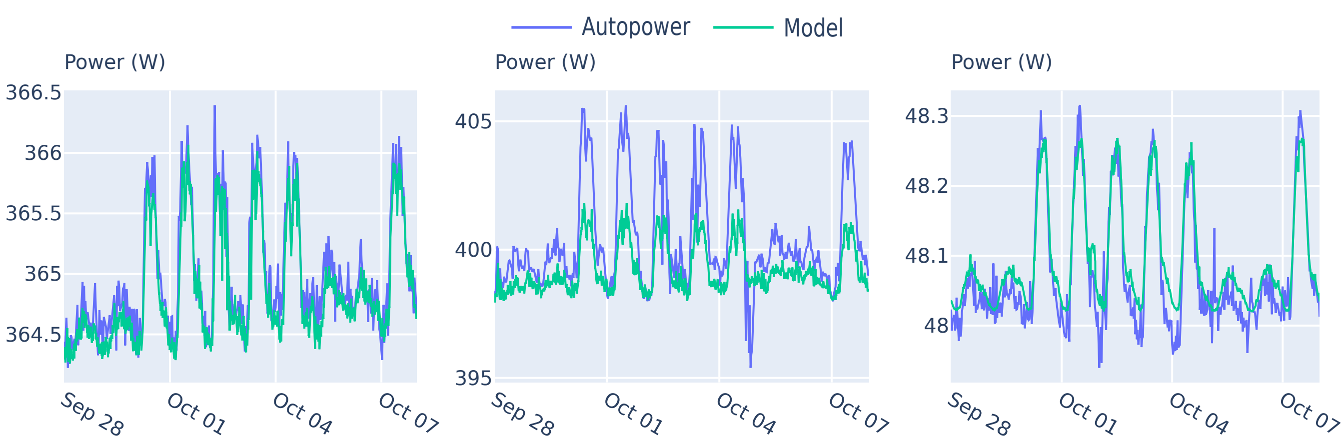


Figure 8: On March 13, an OS upgrade led to an increase in fan speeds, resulting in a 45 W increase (≈ +12%). Nothing changed otherwise. ***Router model:*** **`8201-32FH`**

Table 6: Additional power models derived using the methodology described in § 5 but not explicitly discussed or used in this paper.

	Port	Trans.	Speed	P_{base} [W]	P_{port} [W]	$P_{\text{trx,in}}$ [W]	$P_{\text{trx,up}}$ [W]	E_{bit} [pJ]	E_{pkt} [nJ]	P_{offset} [W]
(a)	EdgeCore `Wedge 100BF-32X`									
	QSPF28	Passive DAC	100G	108	0.88	0	0.69	1.7	7.2	0
	|	|	50G	-	0.21	0	0.31	2.5	5.6	0.05
	|	|	25G	-	0.21	0	0.1	2.7	4.7	0.06
(b)	Cisco `Nexus 93108TC-FX3P`									
	QSFP28	Passive DAC	100G	147	0.17	0.11	0.23	5.4	21.2	0
	QSFP28	Passive DAC	40G	-	0.07	0.11	0.16	6.5	17.4	0.03
	RJ45	T	10G	-	2.06	0.11	0	6.7	16.9	-0.03
	RJ45	T	1G	-	0.93	0.11	0	33.8	18.2	-0.03
(c)	Extreme Switch `VSP-4900`									
	SFP+	T	10G	8.2	0.08	0.06	0	25.6	26.5	0.04
(d)	Cisco `Catalyst 3560`									
	RJ45	T	100M	40	0.21	0	0	15.7	193.1	-0.01



(a) `8201-32FH` (b) `NCS-55A1-24H` (c) `N540X-8Z16G-SYS-A`

Figure 9: A zoomed-in version of Fig. 4 where the power model prediction is manually offset to the level of the Autopower measurements to show the precision of the model predictions.