

From Measurement to Emissions: Assessing the Carbon Footprint of Traffic Flows

SAWSAN EL-ZAHR, University of Oxford, UK

NOA ZILBERMAN, University of Oxford, UK

As sustainability becomes a requirement in network operations, accurately quantifying the carbon footprint of Internet traffic is essential. While energy-aware networking has seen significant attention, the ability to trace carbon emissions at the flow level remains an open challenge due to the complexity of shared infrastructure and lack of related telemetry. In this paper, we present a methodology to obtain fine-grained per-flow carbon emissions from traffic statistics. To this end, we collect power measurements from three switches under varying traffic conditions, including synthetic and real-world traces. From these measurements, we derive a regression model that accurately estimates instantaneous router power consumption using only throughput and packet rate counters—achieving >96% accuracy across all switch types and traces. We then extend this model to compute per-flow carbon emissions, distinguishing between consequential and attributional perspectives, and validate the results using traces from CAIDA and Google services. Our findings uncover actionable insights into how flow and network characteristics such as packet size, packet rate, and network utilization influence carbon cost. Finally, we propose feasible deployment strategies for flow-level carbon estimation frameworks. This work provides a foundational step towards enabling carbon-aware flow-level decision-making for users, applications, and network operators.

1 Introduction

Reducing the carbon footprint of the Internet has become a key objective for major industry players—including Internet Service Providers (ISPs), Content Distribution Networks (CDNs), and content providers—committed to significant emissions reductions by 2030 [4, 7, 13, 33, 51]. Meeting these goals requires more than high-level strategies; it requires a detailed understanding of the carbon impact of specific applications, including their network component which can be of great significance [36, 61].

To enable actionable insights, carbon tracing at the level of individual Internet flows is essential. This involves quantifying the carbon contribution of end-to-end flows as they traverse the network and are processed by end devices. Such fine-grained visibility is critical for designing carbon-efficient systems and applications.

For example, a user or company seeking to align their Internet usage with periods of abundant green energy would need a clear understanding of their actual carbon savings. Additionally, a content delivery company utilizing the networks of multiple ISPs would require an accurate breakdown of its carbon footprint across these providers for sustainability reporting. Therefore, there is a need for carbon tracing at the flow level, both offline for billing purposes, and online to enable real-time application adjustments.

In power systems, tracing energy flows to track carbon sources is already possible [55] and companies use it to weight the greenness of energy consumed at different spatial granularities [12, 47]. In analogy to power systems, carbon tracing for Internet flows is essential in telecom systems.

Unfortunately, obtaining flow level energy and carbon information is difficult because many flows share the same infrastructure and there is a lack of corresponding in-network telemetry. Moreover, the intermittent availability of green energy per region causes carbon metrics to frequently change.

While there is a growing interest in characterizing the energy and carbon footprint of the Internet infrastructure, existing work has primarily focused on data center operations [6, 20, 37],

emphasizing the computing side while overlooking the network component. Efforts in carbon-aware networking have examined optimizing routing [27, 61] and load shifting [28, 36] based on the energy mix, but these did not consider extracting flow-level information, or did not distinguish between the carbon scopes of the flow-level footprint, namely consequential emissions—directly caused by flows—vs. attributional emissions—indirectly linked to the flows and attributed to them for accounting purposes. Similarly, router power models [25, 42, 66] were not extended to the flow-level granularity, and were tested with limited traffic characteristics and router utilization levels. Although standardization initiatives [9, 22, 72] propose high-level flow-based carbon metrics, they remain conceptual, lacking empirical validation or deployment strategies. In brief, to our knowledge, no prior work has empirically linked flow-level traffic statistics to flow carbon emissions, taking into account both attributional and consequential carbon emissions.

In this work, we take the abstract metrics definitions and quantify them through empirical measurements. We focus on monolithic network routers and study how energy and carbon consumption vary with flow characteristics. We focus on hardware application-specific integrated circuits (ASICs) that have energy consumption properties associated with traffic properties such as throughput and packet rate. We are building a model that ties these network properties to the ASIC energy consumption. Since our focus is on the switch ASIC, we use the terms switches and routers interchangeably, as modern commercial devices often serve both functions simultaneously [2] and acknowledging that routers will have an additional software layer on top of the ASIC.

To study the correlation between carbon and flow characteristics, we take two steps: (1) we conduct extensive power measurements and derive a *router*-level model that maps traffic statistics into power values, achieving >96% accuracy. (2) We then extend this model into *flow*-level energy and carbon models enabling routers to compute per-flow emissions using only standard traffic counters, without requiring real-time power measurements.

In summary, we make the following contributions:

- We introduce a methodology to attribute carbon emissions to traffic flows based on traffic statistics.
- Based on an extensive measurement study, we develop a model that converts routers' traffic statistics, read from counters, into power consumption, energy usage and carbon values.
- We extend the router-level model to a flow-level model to attribute carbon emissions to individual traffic flows, and use it to evaluate the emissions of a video streaming use case.
- We propose and compare three methodologies to extract end-to-end flow-level carbon emissions, discussing their accuracy and deployment overheads.
- To facilitate reproducibility and further research, we release our power measurement benchmark and ns-3 simulation code as open-source [29, 30].

The paper is organized as follows: background on carbon scopes and router power models is given in §2, and flow-level carbon tracing in §3. Power measurements and the construction of router- and flow-level models are detailed in §4–6. Deployment scenarios are detailed in §7, a video streaming simulation in §8, followed by discussion (§9), related work (§10), and conclusions (§11).

2 Background

Accounting for carbon emissions involves three scopes: (1) direct emissions, (2) indirect emissions from purchased electricity, and (3) all other indirect emissions in the value chain [49]. The definitions may seem simple, but it is important to note that with shared resources, elements in scopes will vary according to the viewing angle. For instance, an ISP running a network of interconnected routers will account for all the energy consumed by these routers in their scope 2 emissions. However, from an application perspective that is occasionally sending a flow through this ISP, it will only

account for the partial energy consumed by routers that is due to its flow. This will be referred to as **consequential** carbon emissions and falls under scope 2 of the application's carbon footprint. Some may claim that the energy consumed by routers to maintain this connection should also be associated with flows. This will be referred to as **attributional** carbon emissions and, if included, would fall under scope 3 of the application's footprint.

These competing views are the origin of misconceptions and double counting in carbon reporting. A recent conference report on green networking [21] suggests that a definition of metrics is needed while clearly highlighting consequential and attributional carbon metrics and discussing their usefulness in helping applications make informed decisions. In this work, we clearly distinguish between these metrics at the flow level.

The goal of this work is to introduce a methodology for attributing carbon emissions to traffic flows based on traffic statistics. Building benchmarks to characterize system performance is a well-established practice, particularly in the context of evaluating workloads, platforms, and machine learning pipelines [35, 39, 70]. We adopt a one-time micro-benchmarking scheme to build carbon model of traffic flows that vendors can share along with datasheets. This enables a consistent reproducible evaluation of carbon emissions across applications and workloads, bridging benchmarking methodology with sustainability analysis.

In this study, carbon emissions associated with the energy consumption of equipment will be tackled. This can be derived by multiplying this energy consumed by the carbon intensity of this energy. The carbon intensity is the amount of carbon emitted to produce 1 kWh of electricity and is an indicator of the greenness of the energy [12]. To link energy with flows, a power model is needed to understand the dynamics of power with respect to traffic properties. The power measurements in this study are the basis of our analysis.

Existing power models of routers have been investigated in previous works. A summary of the proposed power models is provided below. These categories are illustrated in Fig. 1.

- (1) Constant power consumption independent of utilization [8, 48, 64].
- (2) Fully power proportional with utilization [5, 8].
- (3) Offset + Proportional: Power proportional to the utilization with an additional constant idle power offset [5, 14, 25, 62, 66]. Different types of routers will show different levels of idle and dynamic powers.
- (4) Offset + Multi-step: Power increases in steps with the increased utilization and operating rate [5, 8, 32, 65].

From the literature [25, 46, 66, 67], the idle power P_0 is commonly large ($\sim 90\%$). However, the idle/dynamic power ratio varies with the type of equipment and the efficiency level of a router. The high idle power of the switches is due to the different idle components that are part of the chassis and are turned on all the time. These components include fans, power supply, management processor, and switching fabrics.

In summary, it can be seen that assumptions about the power model vary greatly in the literature because of the evolution of switches with time, as well as the differences in the setups between different switch suppliers.

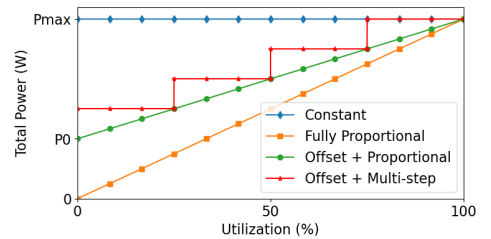


Fig. 1. Power Models of Switches

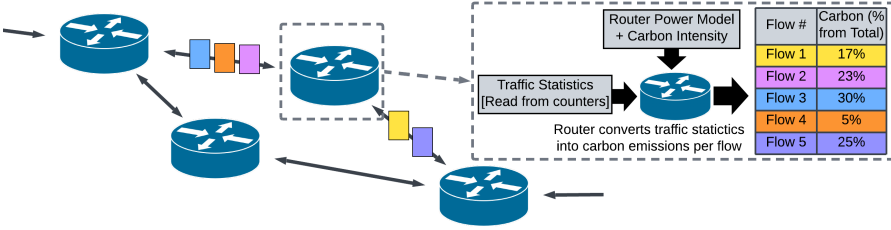


Fig. 2. Flow-Level Carbon Tracing

3 Flow-Level Carbon Tracing

To build energy and carbon traces of flows, a generic power model of a device is not enough. We first need to understand the contribution of each flow to the overall energy consumption and carbon emissions on each device. The power- and carbon-related metrics must be at the flow level.

Figure 2 illustrates the concept of per-flow tracing at the router. It is important to note that we can trace both the energy and carbon contributions of flows. As flows pass through routers along the way, the **power** usage of each router increases accordingly. The resulting **energy** consumption is calculated as the product of this power and the time the flow spends at the router. This distinction is important: power refers to the instantaneous rate of energy use (e.g., watts), while energy measures the total consumption over time (e.g., kWh).

The corresponding **carbon emissions** are then calculated by multiplying the energy consumed by the carbon intensity of the energy source supplying the router. The **carbon intensity**, measured in gCO_2/kWh , is the weighted sum of carbon emissions emitted to produce 1 kWh of energy and is an indication of the greenness of this energy [12]. For example, renewable sources like wind have near-zero carbon intensity, while fossil fuels such as oil can reach up to $935 \text{ gCO}_2/\text{kWh}$ [12]. The lower the carbon intensity, the lower the carbon emissions. Therefore, accurately tracing the carbon footprint of a flow requires knowledge of the carbon intensity at each router through which it passes. The total **carbon trace** is the cumulative sum of emissions generated across all routers along the flow's path, accounting for the carbon intensity of the energy consumed at each point.

It is important to account for both the space- and time-granularity of the carbon intensity metric as it varies significantly between regions and throughout the day. This implies that national average values of the carbon intensity or monthly averages significantly reduce the accuracy of the carbon footprint estimate of flows.

Many flows exist at the same time on the router as described in Fig. 2. Focusing on a given time-window t , the router consumed some energy e_t that had a carbon intensity c_t . The total carbon emissions of the router can be calculated by $e_t \times c_t$. This information is useful for the owner of the router, the ISP, but does not provide useful information to the end user or application. Using traffic statistics of flows on the router such as their throughput and packet rate in the given time-window, the router can infer the percentage of contribution of each flow to the total carbon emissions. The questions to be answered in this paper are: (1) what is a good power model for the router? (2) What flow statistics are necessary to achieve acceptable accuracy? (3) How to communicate the flow-specific carbon trace back to the end user or application?

To answer Question (1), we conduct extensive power measurements on available routers in §4 and compare different power models in §5. We answer Question (2) in §6 and explain how to extend the power model into carbon models at the flow-level. Finally, deployments of the model are discussed in §7 to answer Question (3).

The end-to-end per-flow carbon impact varies based on the transmission path and the type of devices processing the packets. In this work, we focus on the *router* power/energy/carbon modeling and hope to extend it to other network devices.

4 Power Measurements of Switches

4.1 Equipment and Setup

To cover a variety of switches, power measurements were performed on 3 different switches of the same size (32x100G ports), from different vendors, and having different target applications (e.g., low latency, deep buffer, programmability). Switch manufacturers are anonymized due to the license requirements of one of the vendors and will be referred to as 1, 2, and 3. The results of the power measurements were corroborated by the switch manufacturers.

To monitor power consumption, switches are connected to power distribution units (PDUs) as illustrated in Fig. 3. Each power supply unit (PSU) is connected to a different PDU port. The resulting power of the switch is the sum of the power read from each PDU socket. The PDU provides power monitoring with an absolute accuracy of ± 1 W over the full operating range.

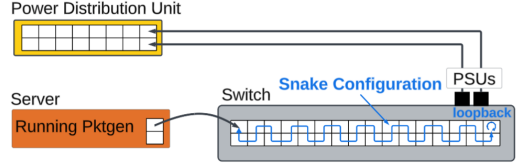


Fig. 3. Experiment Setup

We aim to test each switch under various traffic conditions. A server is connected to one port on each switch and uses Pktgen to generate high-throughput traffic. For each experiment, a pcap file specifying the desired packet size and payload is uploaded to Pktgen, and traffic is sent at a defined throughput. The server connects to the switch via a 100G link.

Two types of pcap files are used: (1) real traffic traces from CAIDA and Google Services, and (2) synthetic traces with packet sizes ranging from 64 to 1500 bytes. The synthetic traces are used to build the power model, which is then tested on the real traffic traces. The use of synthetic traffic traces enabled us to build a balanced power model that avoids bias toward the more common packet sizes typically seen in real traffic.

To generate higher throughput across the switch fabric, we use a snake configuration. In this setup, traffic flows from one port to the next across the switch. Loopbacks at the end of the chain return traffic in the reverse direction, allowing all ports to be fully utilized and driven to 100% capacity. This configuration is essential because sending traffic from a single server alone cannot saturate the entire switch; the snake setup enables us to stress the switch at full throughput.

We evaluated three switches using different forwarding implementations: Switch 1 runs a vendor-supplied reference design, Switch 2 runs VLAN-based forwarding, and Switch 3 runs a commercial protocol stack. Due to differences in architecture, low-level APIs, and available features, it is not feasible to run the same software stack or program across all switches.

Next, we show the power measurements results for the three switches with (1) synthetic traffic traces while varying the throughput and packet size, and (2) real traffic traces while varying the throughput. We compare the power variation of the three switches accordingly.

4.2 Power vs Throughput

The first experiment studies the variation in power consumption when the overall throughput of the switch increases. With the snake configuration, we were able to explore the power variation at nearly 100% throughput.

Figure 4 shows both the measured power from PDUs and the measured throughput from the switch counter while varying the Pktgen rate. Note that the Pktgen rate dictates the bit rate of traffic over the 100G link connecting the server and the switch. Hence, the packets received from the server on the first port of the switch circulate on all 32 ports, resulting in an overall throughput of $32 \times \text{Pktgen Rate}$. This throughput may be reduced due to packet drops when reaching the maximum capacity of the switch. Moreover, due to technical problems, only 26 ports are used for

Switch 2 resulting in a throughput of approximately $26 \times \text{Pktgen Rate}$ for this switch. This only means that a lower overall throughput could be reached on this switch (up to 2.6 Tbps).

The throughput figures show the total throughput of packets processed by the switches. It is important to note that for clarity only the curves for packet sizes of 64B and 1500B are shown but the overall range of packet sizes is used to build the power model in later sections. Moreover, we evaluated different port configurations in the snake setup and observed that power consumption correlates with the switch's total throughput. For example, running at 50% throughput across 32 ports consumed the same amount of power as running at 100% throughput on 16 ports.

The general trend for the three switches is a continuous increase in power while increasing throughput. However, there are significant differences between the slopes for different packet sizes. Switches consume higher power when packet sizes are smaller (64B vs 1500B), given the same overall throughput. Moreover, we see that the maximum throughput achieved for 64B packets is different for the three switches, indicating different packet rate limitations.

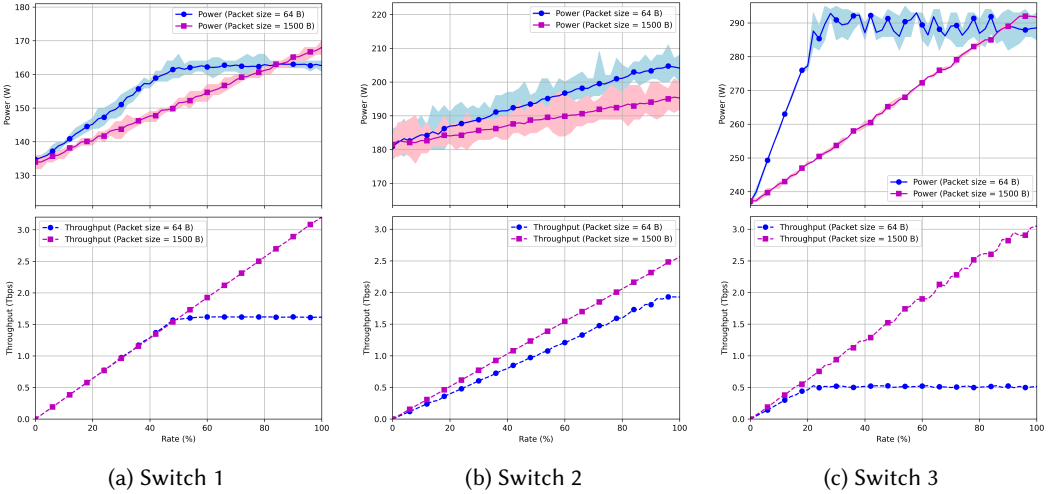


Fig. 4. Switch power and throughput for different input *rates* using synthetic flows

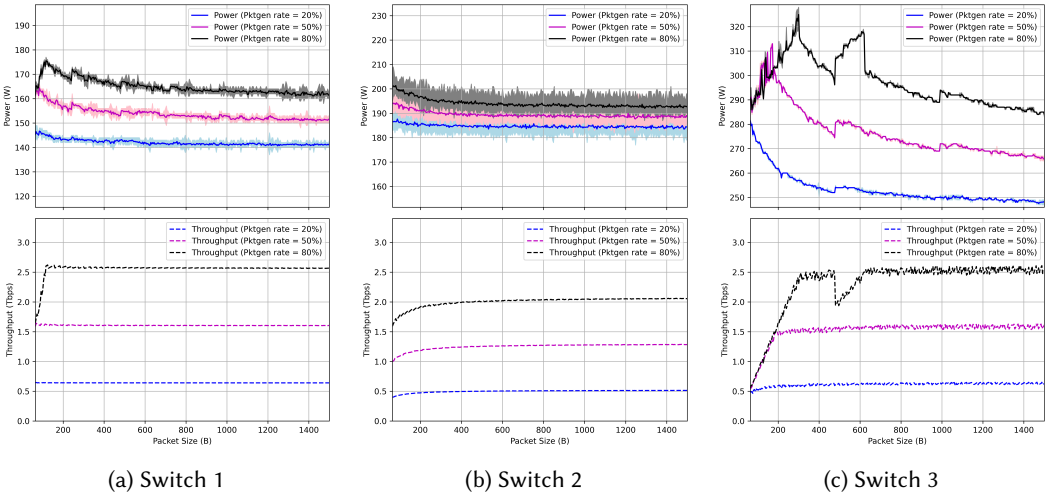


Fig. 5. Switch power and throughput for different input *packet sizes* using synthetic flows

4.3 Power vs Packet Rate

The first experiment in §4.2 only compared between 64B and 1500B packet sizes. This second experiment further studies the power variation of switches when varying the packet size. The full range of packet sizes 64B-1500B is tested with an incremental step of 4B. When fixing the bit rate, the packet rate is inversely proportional to the packet size as expressed in eq. (1).

$$\text{Packet Rate} = (\text{Bit Rate}/8)/\text{Packet Size (including overheads)}. \quad (1)$$

For the same bit rate, packet sizes were varied. Figure 5 shows the measured power of switches at three levels of Pktgen rate: 20%, 50% and 80%. The throughput read from the switch counters is also plotted. Only flows with small packet sizes cannot reach the specified throughput because of packet rate limitations on the switches.

The general trend of power decreasing with larger packet sizes (smaller packet rates) is clear for all throughput levels. However, the scale of power change is different for the three switches. For example, even at 20% utilization of the switch, for Switch 3, packets of size 64B can result in 33W increase in power compared to packets of size 1500B. This power difference is significant with respect to the overall power (13% increase). The power difference at 20% utilization was only 4W for Switch 1 (3%) and 2.2W for Switch 2 (1%).

Another interesting insight is the power jumps at specific packet sizes. This appears slightly at 240B and 440B in Switch 1 but more significantly in Switch 3, where sudden power jumps happen at packets of size 220B, 476B and 988B. These power jumps are significant especially at high throughput values. For example, at 50% utilization of Switch 3, sending packets of size 477B results in a power difference of 6W compared to sending packets of size 476B. 1B difference in packet size resulted in a power jump. Looking more closely at the packet sizes at which power jumps happen, we see a matching between (220B, 476B, 988B) and (256B, 512B, 1024B) with a difference of 36B. These power jumps are likely the result of inefficiencies in the usage of the data bus with a size that matches 256B.

4.4 Idle vs Dynamic Power

The three switches of the same size have different range of power values. Table 1 shows the idle and maximum power values for the three switches along with the percentage of idle and maximum dynamic power with respect to the maximum power. Two observations come from this table: (1) Given that the three switches are of the same size, it is important to notice the big difference between their idle power: for instance, Switch 3 consumes 236W when idle which is 76% higher than Switch 1's idle power (134W). This emphasizes the importance of reducing this idle power consumed regardless of traffic. (2) The idle power ratio with respect to the maximum power is significant for the three switches (72-83%) showing that this ratio is high despite efforts to achieve higher energy efficiency. Compared to idle power ratios from the literature, the range of values of 72-83% is smaller than the reported 80-97% from switches used a decade ago [46, 66]. This indicates that power proportionality of switches improved and still needs further reductions.

Table 1. Comparison between the three switches

Switch	idle (W)	Max (W)	idle/Dyn
1	134	180	74%/26%
2	180	216	83%/17%
3	236	328	72%/28%

4.5 Real Trace Power Measurements

The experiments in §4.2 and §4.3 are based on synthetic data where all packets are of the same size. This enabled us to explore the power variation for a wide range of packet sizes. In this section, real traffic traces are used for measurements and to validate the power models in §5.

Datasets: (1) CAIDA packet trace collected from high-speed monitors on a 10G commercial backbone link in 2019 [1]. This trace has an average packet size of 880B. (2) A dataset of a collection of flows from five Google services [56]: Google Drive, YouTube, Google Docs, Google Search, and Google Music. Table 2 shows the average values of total bytes, total packets, packet size, and flow length for the five traces of Google services.

Figure 6 shows how the power consumption varies between the three switches when the real traffic trace pcaps are replayed through Pktgen at different rates. The results reveal that switch 3 exhibits the highest power variation between the traces, consistent with the trends observed in Figure 5. Notably, the GoogleDoc trace, with an average packet size of 279B, leads to significantly higher power consumption on switches. When comparing the traces, the lower their average packet size, the higher their power contribution. These findings reinforce the power variation patterns discussed in previous sections.

Table 2. Flow Statistics (average values) of Traces Collected from Google Services

Trace	# Bytes	# Pkts	Pkt Size	T(s)
GoogleDoc	0.7 MB	2514	279 B	114.5
GoogleSearch	0.4 MB	666	606 B	10.3
GoogleDrive	11.4 MB	10575	1087 B	32.7
YouTube	19.3 MB	15796	1226 B	56.9
GoogleMusic	8.9 MB	7095	1253 B	3.1

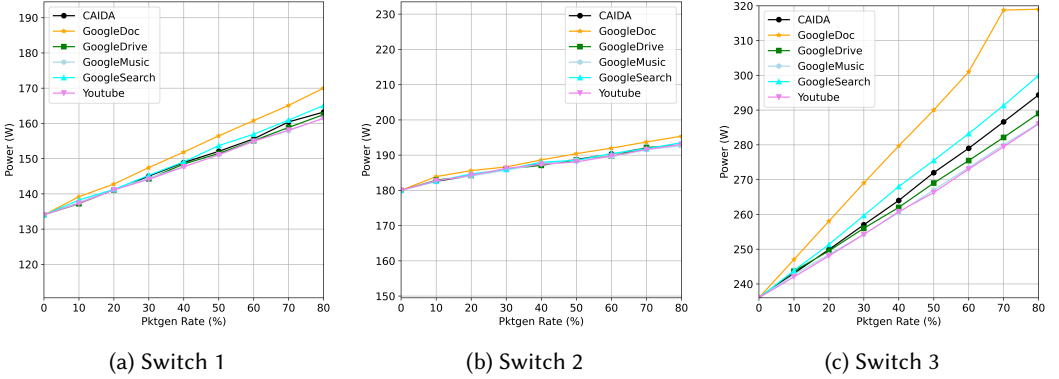


Fig. 6. Switch power measurements for different input rates using *real traffic pcap traces*

5 Power and Carbon Modeling

Accurate estimation of instantaneous power consumption of a switch is important for energy-aware network management, optimization, and monitoring. While this can be measured directly using an embedded power sensor, eliminating the need for external PDUs, a simpler and more cost-effective alternative is to use a power model that estimates power using existing switch counters. Building on the measurements presented in §4, we derive in this section a switch power model based on two key counters: total throughput and packet rate.

5.1 Complexity of the Power Model

In this section, we examine the structure and complexity of the power model. Table 3 reports the coefficient of determination (R^2 score) for various regression models, which indicates how well each model captures the characteristics of the measured power consumption. The power model is trained using synthetic traffic traces composed of single-packet-size flows, and then tested on real traffic traces of CAIDA and Google. Specifically, the test set that includes traces from Google services is aggregated into a single dataset labeled as "Google" in the table. We report the R^2 scores for all three switches, evaluating model performance on both the synthetic data and the more complex real-world traces from CAIDA and Google.

Table 3. Comparison of regression models (in R^2 score) estimating the power model for the three switches. Trained on synthetic single-packet-size traces and tested on real traces from CAIDA and Google Services

Model	Switch 1			Switch 2			Switch 3		
	Single Pkt	CAIDA	Google	Single Pkt	CAIDA	Google	Single Pkt	CAIDA	Google
Linear Regression	99.1	99.8	98.8	98.6	99.1	98.4	96	99.7	96.4
Polynomial Regression	99.4	99.8	99.5	98.8	99	98.4	96.2	99.7	96.7
Lasso Regression	97.2	97.9	97	84	73.6	78.5	95.6	98.8	96.2
ElasticNet Regression	81.1	79.6	81.3	73.9	57.7	64.7	86.3	86.6	89.4
KNN	75.5	91.7	82.9	78.2	59.3	78	89.6	93.2	85.8
Decision Tree, depth=2	92.3	75.4	75.5	90.9	75.3	74.1	77.9	72.9	72.1
Decision Tree, depth=4	97.3	83.3	89.3	97.7	83.4	87.3	96.6	91.8	93.5
Random Forest	99.8	96.6	97.1	99.5	90.5	91.4	99.7	91	94.7
SVM	53.6	94.1	61.6	61.4	81.5	63.8	70.9	97.7	75.7
XGBoost, depth=2	74.7	73.2	75.4	73.7	62.9	63.4	74.3	68.6	70.5
XGBoost, depth=4	93.3	92.8	93	92.4	84.7	85.4	92.9	89	89.1
NN with ReLU	97.6	97.2	96.9	94.1	90.8	92.3	93.8	95.7	93.3
NN without ReLU	94.0	96.5	96.0	91.9	88.1	90.6	93.9	97.8	95.7

Among the models evaluated, linear and polynomial regression achieve the highest performance, with R^2 scores exceeding 91% across all packet traces and all three switches. In contrast, more complex models such as decision trees perform well on synthetic traces with fixed packet sizes but show a significant drop in accuracy when tested on real traffic. This is likely because real traces contain a mix of packet sizes, which smooths out the differences that complex models tend to overfit to in synthetic data. Additionally, the second-order polynomial model offers only a negligible improvement over the linear (first-order) model. Hence, the linear model is not only accurate, but it also outperforms other complex models and will be a low resource to implement.

Table 4. Linear Power Model Parameters

Switch	Idle (W)	α (W/Tbps)	β (W/Mpps)
1	134.1	10.69	0.0054
2	181.6	5.16	0.0038
3	235.9	16.56	0.0441

The linear power model is expressed in equation (2) and its numerical values are summarized in Table 4. The model demonstrates strong predictive performance, achieving an R^2 score between 96% and 99.8%, and a mean absolute error (MAE) ranging from 0.3 to 2.7 W.

$$p = \text{Idle Power} + \alpha \times \text{Throughput} + \beta \times \text{Packet Rate}. \quad (2)$$

To better interpret the significance of the MAE values—especially given the relatively high idle power consumption of switches—we express MAE as a percentage of the mean power, both including and excluding the idle component. The R^2 score remains unchanged in either case, as idle power constitutes a constant offset in the model. When normalized to the mean total power (including idle power), the MAE ranges from 0.17% to 0.8% whereas when normalized to the mean dynamic power (i.e., mean power minus idle power), the MAE is 1.8–5.5%.

5.2 Power Model Features

The features used in the model are the throughput and the packet rate read from existing counters. Many previous works rely solely on the throughput values for their models [27, 36, 61, 63, 67].

Table 5. Linear model accuracy with different features. Higher R^2 and lower MAE indicate better performance. MAE-T and MAE-D are normalized MAE percentages including and excluding idle power, respectively.

Switch	Features	R^2 Score (%)	MAE (W)	MAE-T (%)	MAE-D (%)
1	Throughput Only	90.6-96.9	1.4-2.2	0.9-1.4	9.1-12.2
	Pkt Rate Only	22.3-31.6	6.4-7.0	4.2-4.5	154.5-439.5
	Throughput and Pkt Rate	98.8-99.8	0.3-0.8	0.2-0.5	1.8-4.9
2	Throughput Only	75.6-95.8	0.8-1.3	0.4-0.7	10.8-17.9
	Pkt Rate Only	15.7-46.6	2.7-3.3	1.4-1.8	131.3-151.6
	Throughput and Pkt Rate	98.4-99.1	0.3-0.4	0.17-0.2	4.0-5.5
3	Throughput Only	52.9-78.6	6.9-10.9	2.6-4.0	35.5-49.3
	Pkt Rate Only	49.9-66.3	9.8-11.3	3.6-4.1	63.1-100.4
	Throughput and Pkt Rate	96.0-99.7	0.7-2.1	0.3-0.8	2.1-5.4

Table 5 compares the accuracy of the linear power model when using different input features: (1) throughput only, (2) packet rate only, and (3) both throughput and packet rate. The evaluation metrics include the R^2 score, mean absolute error (MAE) in watts, and MAE as a percentage of the mean power—calculated both including and excluding idle power. The models are evaluated on three datasets: synthetic traffic, CAIDA traces, and Google traces.

The results indicate that using throughput alone substantially reduces model accuracy, particularly for Switch 3, where the R^2 score drops to 52.9–78.6% compared to using both features. Relying solely on packet rate results in even lower accuracy across all switches and datasets. Notably, Switch 3 benefits the most from using both features; it performs poorly when either feature is used individually, unlike other switches, which achieve reasonable accuracy using throughput alone. Incorporating both throughput and packet rate significantly improves performance, increasing the R^2 score to 96–99.7% and reducing the MAE percentage (excluding idle power) from 35.5-100.4% to below 5.5%.

These findings highlight the importance of using both features to achieve high model accuracy, with R^2 scores exceeding 96% and MAE percentages <6%.

5.3 From Power to Energy to Carbon Model

The carbon emissions associated with a router are the product of its energy consumption (cumulative power over a time window $[t_1 : t_2]$) and the carbon intensity of this energy. This can be expressed as $\int_{t_1}^{t_2} (p_t \times c_t) dt$. The accuracy of this carbon model has the same accuracy as that of the previously derived power model.

6 Carbon Model at the Flow Level

The power model developed in §5 is for the total power of the switch based on the switch counters. The goal of this work is to dive further and derive an energy/carbon model of the traffic flows: given the per-flow traffic properties, what is the energy/carbon contribution of each flow? We separate (1) consequential carbon emissions of the flow, which are the carbon emissions directly caused by the processing of the given flow and thus are related to the dynamic power of the switch, and (2) attributional carbon emissions of the flow that are not directly caused by the flow, but are related to its contribution to the idle power of the switch. Note that only operational carbon emissions are in scope of this study, thus, embodied carbon emissions—the carbon due to *manufacturing* of

devices- that can also be attributed to flows are not included in this analysis but can be equivalently disseminated in a similar fashion.

6.1 Consequential Energy and Carbon Emissions of Flows

When considering traffic flows, we need to look at the cumulative power that this flow contributed to over the flow duration. This means that we need to move from the instantaneous power model in §5 to an energy model per flow. In this section, we look at the dynamic part of the power model developed in §5, that is $p = \alpha \times \text{Throughput} + \beta \times \text{Pkt Rate}$. The contribution of flows to idle power is discussed as part of attributional energy/carbon per flow in §6.2.

The first step is to detail the characteristics attributed to each flow. A flow is typically defined by a 5-tuple key: IP source and destination addresses, transport-layer source and destination ports and IP protocol. The attributes per flow include, among others, the cumulative packets (denoted by p_f) and bytes (denoted by b_f) counters and the flow starting and finishing timestamps. The flow duration will be denoted by t_f .

Applying the linear power model from eq. (2) in §5 to the flow characteristics, and only looking at the dynamic part over the flow duration, that is directly caused by the flow, we get the energy per flow, denoted by e_f formulated in eq. (3).

$$e_f = \alpha \times \frac{b_f \times 8}{t_f} \times t_f + \beta \times \frac{p_f}{t_f} \times t_f = (\alpha \times 8) \times b_f + \beta \times p_f. \quad (3)$$

This implies that the values extracted from the power measurements modeling α and β previously expressed in W/bps and W/ppts, can be used for the flow energy modeling as $\alpha' = \alpha \times 8$ in J/B and $\beta' = \beta$ in J/packet. With correct scaling and multiplying by the carbon intensity of energy, the resulting carbon metrics are expressed in gCO₂/B and gCO₂/packet.

We convert the power measurements in Fig. 6 to energy measurements per flow by normalizing by the number of flows per trace (given all flows are of the same category per trace) and by taking into account the average flow duration in Table 2, we get the average *consequential* energy consumption per flow for the 5 Google Services datasets, as illustrated in Fig. 7. The predictions based on the linear energy model compared to the measurements for the flows in the Google traces, have a mean absolute percentage error of 3-7%, for the three switches.

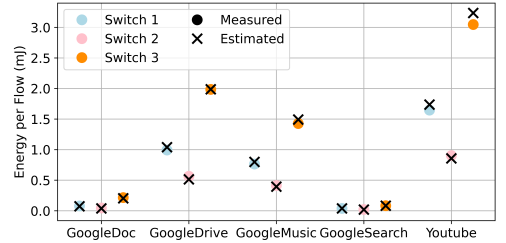


Fig. 7. Measured vs estimated average (consequential) energy per flow for traces from Google services.

6.2 Attributional Energy and Carbon Emissions of Flows

The idle power of switches remains constant regardless of their utilization. In other words, when flows traverse a switch, only the dynamic power varies with traffic load, while the idle power—which constitutes a significant portion of the total (72–83%)—remains unchanged. As such, the idle power cannot be attributed to individual flows as scope 2 emissions, but rather should be included under scope 3 emissions. This assumption is based on the hypothesis that the switches are powered on to enable the forwarding of network flows.

This hypothesis raises two key questions: (1) What are the possible definitions of the flow's contribution to idle power? and (2) What are the implications of these definitions when the flow or network characteristics change?

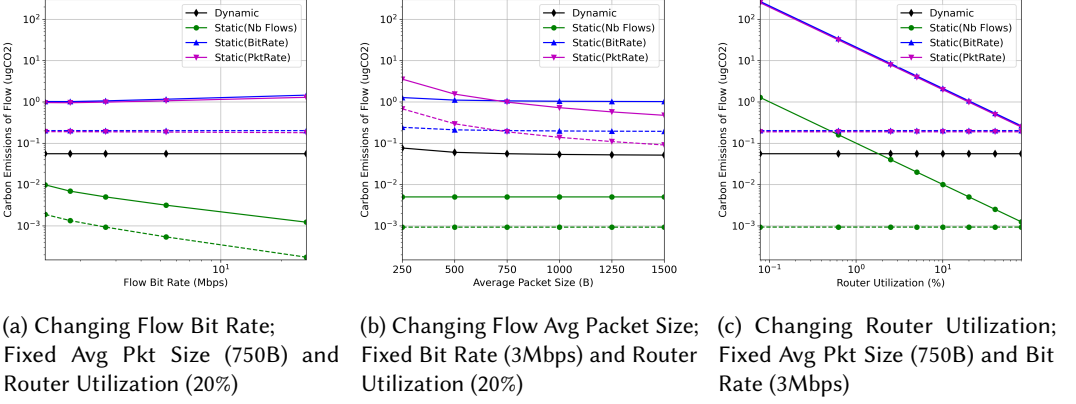


Fig. 8. Change in carbon emissions for the given flow as we vary the flow and network characteristics. Dashed lines represent attributional carbon emissions accounting for router utilization.

6.2.1 Idle Power Attribution Models. We introduce four definitions for attributing idle power or energy to network flows:

- (1) **Zero idle power:** flows do not contribute to idle power.
- (2) **Equal division by number of flows:** the idle power is evenly divided among all active flows in a given time window.
- (3) **Division based on bit rate:** the flow's share of idle power is proportional to its bit rate, attributing more idle power to higher-throughput flows.
- (4) **Division based on packet rate:** similar to the previous definition, but idle power is attributed based on the flow's packet rate.

It is important to note that the contribution of flows to idle power is not directly measurable but is rather attributed based on the adopted definition.

6.2.2 Extensions Based on Utilization. An additional consideration is whether the total idle power should be divided across all flows, or only a portion of it corresponding to the utilization of the switch. This results in three additional cases extending definitions (2)-(4), where the idle power assigned to flows is scaled proportionally to the measured utilization of the switch. The implications of these definitions are best illustrated through an example.

6.2.3 Illustrative Example. To demonstrate the impact of various transmission parameters on carbon attribution, we consider the case of sending a 10 MB file over a network. This example assumes the power characteristics of Switch 1 and a baseline switch utilization of 20%. The file is transmitted on top of background traffic derived from CAIDA traces (scaled to maintain the target utilization), with an average carbon intensity of 200 gCO₂/kWh. The analysis spans a one-minute interval, with flow samples taken every second.

Effect of Flow Bit Rate. Figure 8a shows how varying the flow's bit rate (with a chosen fixed packet size of 750 B) influences carbon emissions. A higher bit rate shortens the flow's duration. Under attribution schemes where idle power is evenly divided among flows, shorter flows receive less idle energy allocation—thus lowering their attributed emissions. With per-second sampling, each active flow is allocated emissions in every slot it traverses the switch. Hence, transmitting the same file faster shortens its active time and reduces attributed emissions “Static(Nb Flows)”. However, if idle power is attributed based on bit rate or packet rate, increasing the bit rate can cause

the flow to be categorized as a “heavy hitter,” which increases its share of the carbon footprint. The consequential carbon emissions of the flow (related to the switch dynamic power) does not change while varying the flow bit rate. If only the idle power relative to the router utilization is divided among flows (dashed lines in Fig. 8a), the curves follow the same trends however, the scale is approximately 1 order of magnitude smaller compared to dividing the *total* idle power over flows.

Effect of Packet Size. Figure 8b illustrates the impact of changing the packet size while maintaining a chosen fixed bit rate of approximately 3 Mbps. In schemes that attribute idle power based on packet rate, increasing the packet size reduces the number of packets per second. This, in turn, lowers the idle power attributed to the flow, resulting in reduced carbon emissions. The consequential carbon emissions associated with the data flow are slightly affected by changes in packet size. Using smaller packets to send the same file increases the total number of packets, which adds overhead from packet headers. This results in higher overall throughput and slightly higher consequential carbon emissions. Dashed lines accounting for router utilization follow the same analysis as Fig. 8a.

Effect of Network Utilization. Traffic varies throughout the day, peaking around 8 PM [23]. This is an external factor that cannot be controlled by the sending application but still affects the attributed carbon emissions to flows. Figure 8c explores how the router’s utilization impacts carbon attribution. The chosen flow’s configuration remains constant (3 Mbps bit rate and 750 B packet size), while the number of concurrent flows on the router changes to simulate varying network loads.

At peak, the number of flows passing through the router will be high. If the total idle power was divided among these flows, the contribution of each flow to the idle power would be smaller. This is illustrated in the sharp decline in the carbon of flows when the *total* idle power is divided among flows. However, when utilization of the router is considered (dashed lines in Fig. 8c), the flow’s attributed emissions remain relatively stable regardless of network utilization.

6.2.4 Summary and Insights. These examples highlight the influence of idle power attribution models on the resulting carbon emissions of network flows. Depending on the chosen model, different and sometimes contradictory insights emerge: (1) When idle power is evenly divided across flows, transmitting over shorter durations leads to lower attributed emissions. (2) When idle power is attributed based on bit rate, lower bit rates yield lower emissions. (3) When packet rate is used for attribution, larger packet sizes result in lower emissions. (4) When idle power is divided without accounting for router utilization, sending flows during high-utilization periods (e.g., peak hours) reduces their carbon footprint which may incentivize sending more flows at peak.

6.2.5 Toward Standardized Attribution. These findings reveal the ambiguity and sensitivity of attributional carbon accounting in networks. The choice of idle power attribution model can significantly affect the perceived emissions of flows and consequently drive different optimization strategies. Given this variability, it is essential that Internet Service Providers (ISPs), cloud operators, and application developers agree on a consistent and transparent definition of how emissions—particularly from idle power—are attributed. The definitions and examples provided in this section aim to guide such discussions and support ongoing efforts toward standardizing the accounting of carbon emissions in networked systems.

6.3 Metrics and Flow Counters

The flow-level carbon model with both consequential and attributional carbon emissions requires accessibility to flow counters on the router or in a central monitoring unit.

For consequential carbon emissions, using the carbon per byte and carbon per packet metrics, the counters needed are only for the total number of bytes and packets per flow. An ISP can be monitoring specific flows to reduce the size of the table that stores flows’ attributes.

However, to derive attributional carbon emissions per flow, the counters needed depend on the definition chosen. If the idle power is to be divided by the number of flows, then a counter is needed to keep track of the number of flows passing through the router for a given time-window. If the idle power is divided based on the flow bit rate or flow packet rate, the existing counters that read the total throughput and packet rate of the router are needed. If only a portion of the idle power based on utilization level is to be divided among flows then, the router utilization (ratio of total throughput and the maximum capacity of the router ($C_{\max}$)) should be known to the router. Table 8 in Appendix A details the counters and metrics needed per definition.

One special case for attributional carbon emissions is when dividing the proportional idle power over flows based on their bit rate. The formula in this case is represented by:

$$\begin{aligned}
 e_f &= p \times t_f \\
 &= \left(\frac{\text{Flow Bytes}}{\text{Total Bytes}} \times \text{idle Power} \times \text{Router Utilization} \right) \times t_f \\
 &= \left(\frac{\text{Flow Bytes}}{\text{Total Bytes}} \times \text{idle Power} \times \frac{\text{Total Bytes} \times 8/t_f}{C_{\max}} \right) \times t_f \\
 &= \text{Flow Bytes} \times \frac{\text{idle Power} \times 8}{C_{\max}}.
 \end{aligned} \tag{4}$$

Similar to eq. (3), a new *constant* metric $\gamma = \text{idle Power} \times 8/C_{\max}$ of unit J/B can be used in this case. Multiple works on carbon-aware routing [36, 61] used this metric for the ease of its integration. However, this metric is not standardized and it is not agreed on whether it is fair to add it to consequential carbon metrics (α and β). A further analysis of the scale of attributional carbon emissions is presented in §8 in the context of a large-scale simulation.

7 Deployments of the Carbon Model

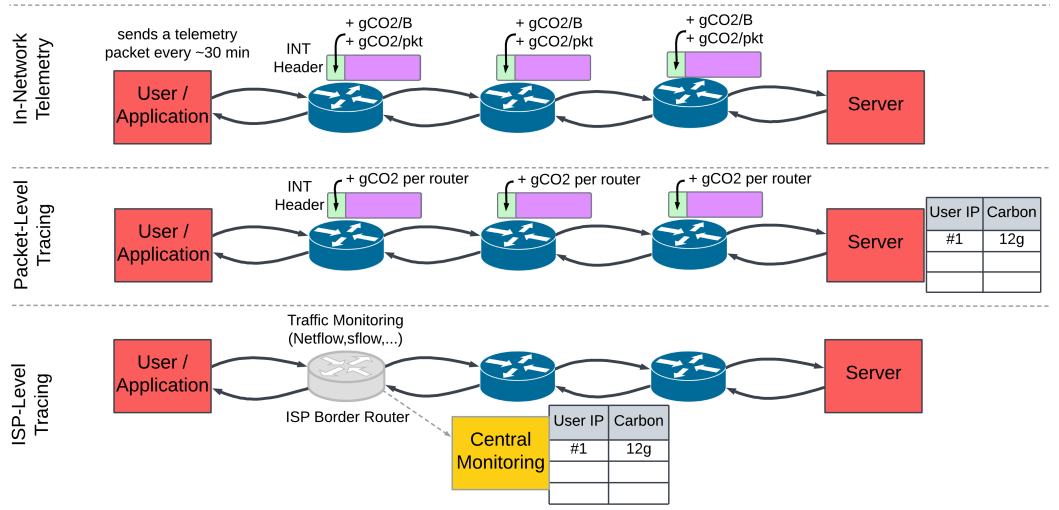


Fig. 9. Carbon Tracing Methods

The goal of this work is to suggest a methodology to attribute carbon emissions to traffic flows based on traffic properties. In previous sections, we derived the per flow carbon model at the level of a single router. To achieve end-to-end carbon tracing, we present in this section three possible

methods, illustrated in Figure 9: (1) using in-network telemetry, (2) using packet-level tracing and (3) using ISP-level tracing. The goal of these methods is to enable applications/users to (1) accurately derive their carbon footprint and (2) adjust routing/scheduling to greener paths or times.

(1) In-Network Telemetry: In this method, the user or application periodically sends a telemetry packet through the network. A new telemetry packet is needed whenever the carbon intensity changes (typically every 30 to 60 minutes [12, 47]) or when network utilization experiences significant variations. Although utilization can fluctuate rapidly, its average behavior over time is generally predictable [27]. The telemetry packet should carry per-packet or per-byte carbon emissions for consequential carbon accounting, and the carbon emissions from idle power divided by the average number of flows, bytes, or packets for attributional accounting.

Upon arrival at the destination server, the telemetry packet is processed, and the collected carbon data accumulated over the path is returned to the sending user/application. Each flow on a different path will get different aggregate carbon telemetry. Increasing the telemetry frequency enhances the accuracy of attributional accounting. This approach enables applications to locally estimate the carbon footprint of the flows they specifically send and potentially adjust their behavior—such as modifying transfer rates or scheduling—to reduce emissions. Given the predictability of carbon intensity, future carbon metrics may also be communicated to users for proactive adjustments.

(2) Packet-Level Tracing: This method provides higher accuracy by embedding an in-band network telemetry (INT) header in every sent packet. Each router along the path updates the packet with its corresponding carbon contribution. The destination server accumulates the emissions for each flow and can report them either to the user upon flow completion or periodically. This approach can be applied in both directions, user to server and server to user.

(3) ISP-Level Tracing: This method leverages existing ISP monitoring infrastructure and requires no changes to packet headers. Border routers and other monitoring nodes already collect traffic statistics for billing purposes. These nodes record flow metadata—such as source/destination IPs, ports, and timestamps—and forward it to a central node. The central node estimates the paths taken by the flows and correlates them with the carbon intensity values of routers along those paths. As a result, the ISP can estimate emissions per flow, user, or cache and report them periodically (e.g., hourly, daily, or weekly). Although packet sampling can reduce accuracy, biased monitoring of selected flows can mitigate this limitation.

Comparison: Each of the three methods presents different trade-offs in terms of accuracy, overhead, carbon data distribution, and update frequency. Table 6 summarizes the comparison.

In terms of *accuracy*, in-network telemetry may be limited by multi-path routing (e.g., ECMP), which introduces uncertainty in flow paths. Packet-level tracing is the most accurate since it captures emissions on a per-hop basis with full path visibility. ISP-level tracing achieves moderate to high accuracy depending on the quality of path estimation and sampling.

With respect to *packet overhead*, in-network telemetry introduces minimal overhead by sending dedicated telemetry packets or occasionally attaching INT headers to flow packets. Packet-level tracing has higher overhead as every packet carries an INT header. The INT header has a size of 12B [34] which is fixed in this case because carbon metrics data is cumulative. With an average packet size of 880B (CAIDA trace [1]), the packet overhead is 1.3%. The INT header can be added at line rate with today's programmable devices that support INT. ISP-level tracing imposes no additional packet overhead, as it relies solely on existing ISP traffic logs. While increasing telemetry frequency improves accuracy, it also slightly increases traffic and associated emissions. However, this trade-off does not eliminate the need for accurate carbon reporting that is a requirement to enable carbon-aware networks.

Table 6. Comparison of Carbon Tracing Methods

Aspect	In-Network Telemetry	Packet-Level Tracing	ISP-Level Tracing
Accuracy	Moderate. Can be reduced by multi-path routing (e.g., ECMP).	High. Full path-level granularity.	Moderate-High. Based on path estimation and sampling granularity.
Packet Over-head	Low. Only telemetry packets carry the INT header.	High. Every packet carries and updates the INT header.	None.
Carbon Data Distribution	Required at each router for local computation.	Required at each router for local computation.	Only required at central ISP monitoring node.
Reporting Frequency	Sent whenever carbon intensity or network load changes significantly.	Sent at flow completion or batched periodically.	Sent less frequently (e.g., daily/weekly/monthly).
Application Adaptability	Enables real-time carbon aware adaptation.	Potential for adaptation with added complexity.	Good for reporting, not dynamic adaptation.

For *carbon intensity data distribution*, in-network telemetry and packet-level tracing require each router to have local access to up-to-date carbon intensity. In contrast, ISP-level tracing centralizes this information, requiring only the central monitoring node to maintain carbon intensity data.

Finally, in terms of *reporting frequency*, in-network telemetry requires updates at flow initiation, when carbon intensity changes and when network load changes significantly. Packet-level tracing allows reporting at flow completion or at low periodic intervals. ISP-level tracing supports only coarse-grained reporting intervals (e.g., daily, monthly), which limits its ability to support real-time carbon-aware application adaptation, although it remains effective for accurate carbon reporting.

8 Video Streaming Use Case

To illustrate the magnitude of per flow carbon emissions, we simulate video streaming workloads on a large-scale real network topology using the ns-3 simulator. The aim is to measure the carbon footprint per flow rather than looking only on the overall router or ISP carbon emissions. This complements router power measurements and offers insights into (1) understanding the flow mix on routers at different levels of a network (core vs edge) which impacts attributional carbon emissions of flows, and (2) tracing of flows on a large ISP network with regionally varying carbon intensity.

Topology: The simulation uses the British Telecom (BT) core network, consisting of 1008 nodes and 3111 bidirectional optical links (up to 100 Gbps). Nodes are hierarchically organized into 20 core, 63 metro, and 925 tier-1 nodes [53]. The access network managed by Openreach (a subsidiary of BT) is excluded. In the context of video streaming, embedded caches used by content providers to serve content locally from ISPs are co-located with metro nodes in the BT topology [27], so traffic flows from caches to users across the network. The topology is illustrated in Fig. 10.

Video Streaming Flows: Users can watch videos in different qualities. We take the Netflix minimum required bit rates for the three video qualities: 3 Mbps for High Definition (HD), 5 Mbps for Full

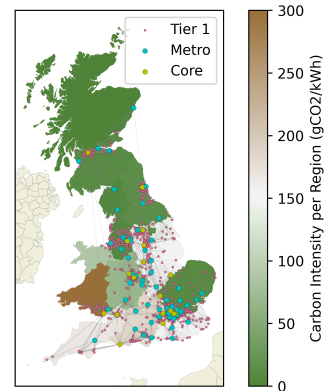


Fig. 10. BT Topology [53]

HD (FHD), and 15 Mbps for Ultra HD (UHD) [52]. We assume that users are evenly distributed across video quality levels. Since higher quality streams require higher bit rates, users watching at higher quality will generate flows with more total bytes and packets over the same viewing duration. The paths taken by flows are different depending on the location of users, and we assume that users are served from one of the two closest caches in the network. The packet size used for these flows is 1500B which is typical for the case of video traffic, where the content is large in size [11]. The simulation includes 300,000 concurrent viewers spread equally among all nodes that make up a total throughput of 2.4 Tbps. This represents 8% of the peak traffic of BT [41] and 50% of the peak Netflix traffic seen on BT [59].

Carbon Intensity Data: The regional carbon intensity in the UK is extracted from [12] for the peak hour of 8PM on 01-01-2025. The UK is divided into 14 regions and using power flow analysis and real telemetry, the carbon intensity of the energy consumed per region was derived [12].

Routers Power Model: Due to the limited availability of publicly documented power specifications for the routers used in the BT topology, we adopt the three linear power models of the three switches that we derived in previous sections. These models are used to approximate router power consumption in our simulations, providing a realistic evaluation scenario. The three models are assigned uniformly across the 1,008 nodes in the BT topology.

8.1 Implementation

The BT topology is implemented in ns-3, where downstream flows at various quality levels are simulated across the network. To estimate the carbon footprint of each flow, we track its path through the network in ns-3 and record the local utilization at each router it traverses. The previously derived models are then leveraged to estimate the energy and carbon footprint-both consequential and attributional-of each flow.

8.2 Results

Impact of Video Quality: Streaming the same video at different qualities results in the use of a different bit rate over the same time duration. This should not be confused with sending the same file size at a higher bit rate because higher quality of video means that a larger file size is transmitted during the same duration. Figures 11 and 12 show the cumulative distribution function of consequential and attributional carbon emissions per flow, respectively. Streaming with a higher bit rate results in higher consequential and attributional carbon emissions per flow. This is true for the attributional carbon emissions based on the bit rate. However, when the total idle part of carbon emissions is attributed to flows equally, the three curves for the three video qualities align together (as shown in Fig. 12).

Scale of Carbon Emissions: Figure 13 shows a box plot of the different scopes of carbon emissions per flow. These values are for flows of the same video quality at a bit rate of 5 Mbps, allowing for a fair comparison between carbon scopes independent of bit rate.

In this large-scale simulation, the consequential carbon emissions of the flows are at least an order of magnitude lower than attributional carbon emissions. Among variants of attributional carbon emissions, incorporating the router's utilization reduces per-flow emissions by approximately two orders of magnitude.

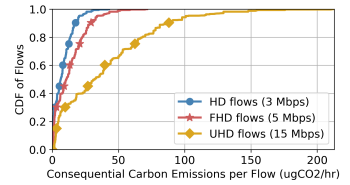


Fig. 11. Consequential carbon per flow for same watching duration

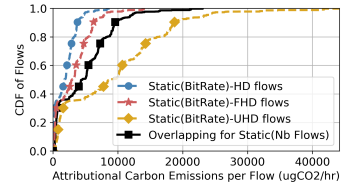


Fig. 12. Attributional carbon per flow for same watching duration

Impact of Network Utilization: BT’s peak end-to-end traffic of 30.1 Tbps implies approximately 30 Gbps downstreaming traffic on the 1008 nodes of BT which is well below router capacity. This substantial overprovisioning leads to significant variance in the per-flow carbon attribution when accounting for utilization. In Fig. 13, carbon emissions per flow are reduced by around 2 orders of magnitude only by accounting for router utilization. This highlights the core dilemma in energy attribution: whether to assign idle power entirely to flows or proportionally to actual load. This simulation highlights the impact of low router utilization seen in ISPs on attributional emissions.

Energy per Byte and Carbon per Byte: The metrics of energy per byte and carbon per byte are often used when projecting the future energy consumption and carbon footprint of networks. In Table 7, we normalize the energy consumption and carbon emissions of the flows to the number of bytes sent per flow. We distinguish between the different scopes of energy and carbon to show the different scales. Interestingly, none of the values of energy per byte and carbon per byte are constant. This shows that the same flows, taking different paths, will have different consequential and attributional energy and carbon. More interestingly, the variability in the carbon per byte metric is higher than in the energy per byte metric (higher relative standard deviation). This is due to regional carbon intensity variation along flow paths (i.e., in the UK at the given simulation time, the carbon intensity between regions had an average of 102 gCO₂/kWh and a standard deviation of 92 gCO₂/kWh). This increases variability in the carbon attributed to flows, in addition to differences in equipment type and hop count.

Impact: A previous Netflix report estimates that streaming one hour of video in Europe emits about 55gCO₂e, with over 50% attributed to the user’s device [50]. Our work focuses on the core network component. For instance, using the BT topology simulation, streaming UHD video for one hour in the UK results in 20.25 gCO₂ in consequential emissions—roughly consistent with Netflix’s figure. However, attributional emissions vary widely, from 101 gCO₂ to 6,642 gCO₂ per hour, depending on the chosen methodology, up to 120 times higher. This highlights the urgent need for standardized definitions to ensure consistent and comparable reporting.

9 Discussion

Power, energy and carbon metrics. On the router level, the power model developed resulted in three power metrics: (1) idle power of router, (2) W/bps and (3) W/pps. The idle power is based on router configuration while the other two metrics are based on traffic properties. Focusing on traffic-related metrics, the power metrics are converted into energy metrics of the form J/B and J/Package. Multiplied by the carbon intensity of the energy consumed, the resulting metrics are of the form gCO₂/B and gCO₂/Package. These are convenient linear metrics that help understand the scale of the carbon footprint directly related to traffic. Operators, before communicating these metrics, should multiply them by their power usage effectiveness (PUE) which is part of the scope 2 emissions of flows and contributes to greater variability between sites.

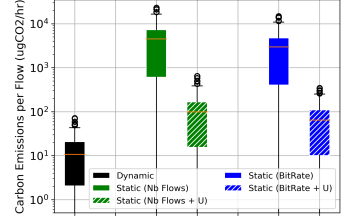


Fig. 13. Carbon emissions scale for FHD flows (5 Mbps)

Table 7. Energy and Carbon per Byte in BT

Metric	Energy (mJ/B)			Carbon (ugCO ₂ /B)		
	Avg	Std	P99	Avg	Std	P99
Dynamic	0.102	0.036	0.176	0.003	0.002	0.01
Idle (Nb Flows)	36.5	23.3	84.2	0.984	1.118	4.485
Idle (Nb Flows + U)	0.841	0.507	1.93	0.023	0.025	0.104
Idle (BitRate)	23.9	6.7	33.5	0.643	0.562	2.279
Idle (BitRate + U)	0.551	0.117	0.755	0.015	0.012	0.055

Accuracy of power modeling. Surprisingly, the linear power model performed better than other complex models especially, for generalization to real traces. The accuracy of this model has >96% R2 score for both router-level and flow-level power modeling.

Packet rate is as important as throughput. Previous work focused on throughput-related metrics. Packet rate was considered to have a small impact on the power. However, this study shows that packet rate is as important as throughput to enhance the accuracy of the power model. These two features are complementary and should be added together to boost the accuracy of power prediction.

Flow-level carbon tracing deployment. Carbon tracing at the flow level is possible and can be done based on existing router counters and monitoring tools. With the use of metrics introduced in this study, tracing can be performed with high accuracy. The suggested deployment solutions allow applications/users to (1) derive their carbon footprint accurately and (2) adapt their routing/scheduling to greener paths or time intervals.

Consequential vs attributional carbon emissions. When quantifying the carbon savings of any approach, it is important to differentiate between consequential and attributional carbon emissions of flows. The debate around the different definitions of attributional carbon emissions should be resolved between ISPs, content providers, and other key stakeholders. This study showed the different scales of attributional carbon emissions. Despite the improvement in power proportionality of routers, the idle power still makes up a large portion of power, which in turn makes the attributional carbon emissions of flows significant.

Same flows, different paths, different carbon footprint. Quantifying the carbon footprint of workloads in datacenters has been widely investigated in previous work, however, the network part is often overlooked. This work helps to quantify the carbon footprint of traffic flows. The same application, when run in different locations, may retrieve data from caches or datacenters through different network paths. These varying paths involve different numbers and types of routers, as well as varying energy sources, all of which influence the overall carbon footprint of the flow and can lead up to 470% difference in emissions (Section 8).

Packet size effect. Using larger packet sizes in flows reduces the packet rate and consequently the power. However, power peaks occur at specific packet sizes with maximum mismatch with the bus width of routers. This is also well known to also degrade throughput [71].

Methodology for power modeling. The power measurements and modeling in this study emphasize the need for vendors to report power consumption across a range of packet sizes and utilization. The methodology should be standardized, and the metrics should be included in equipment datasheets. Alternatively, vendors may share optimized power models of their equipment.

Limitations. Our experiments are based on three monolithic routers, but with significant architectural diversity. Broader validation across other router classes, configurations and topologies remains future work. Our results nonetheless point to consistent and reproducible relationships between traffic and energy, establishing an empirical foundation for further standardization of power models and carbon-aware networking.

10 Related Work

Several studies have examined energy and carbon efficiency in computing. The work in [38] compares energy- and carbon-efficient strategies in cloud systems, showing that optimizing for energy alone can be suboptimal. A carbon-aware scheduler for delay-tolerant batch jobs is proposed in [10]. The work in [45] explores the potential carbon savings by allowing longer internet service response time based on user preferences.

In terms of metrics, an old survey of sustainability metrics for distributed cloud networks was done in [58]. Metrics including operational and embodied carbon per job were proposed in [31],

which aim to measure the end-to-end sustainability footprint in data centers. The work in [26] extends the metrics in [31] to AI-specific sustainability metrics with focus on the entire life cycle.

In terms of modeling, end-to-end energy tracing of requests in a datacenter was investigated in [6] that suggested a GNN model to estimate energy consumption. Training of this model is based on energy measurements API exposed by services used by the request. Similarly, the authors in [20] introduced an ML method to approximate the cost of a data center execution for demand response given the data center configuration (e.g., power, performance, and load characteristics). Carbon modeling for workloads running on a data center server was studied in [37] using a game-theoretic model. At the Linux-level, the work in [54] suggested using eBPF functions to efficiently measure per-process energy consumption at millisecond-scale granularity. In [60], a carbon-aware Linux-based framework was suggested to track carbon down to the level of requests.

In the context of networking, carbon tracing is less investigated. An illustration of carbon-aware networks was introduced in [72] and highlighted the need for in-network telemetry at the flow level to be able to tie carbon to applications and users. Energy-efficient and carbon-efficient approaches are suggested in the context of routing: energy-aware routing [3, 16–19, 24, 43, 64, 68, 69] and carbon-aware routing [15, 27, 36, 44, 48, 57, 61, 63, 67].

Initial modeling works for router’s power consumption include [25, 42, 66]. Power measurements were conducted for multiple routers, but were limited to small traffic rates (up to 200 Gbps) and few packet sizes. A unified method combining bottom-up and top-down energy estimation was proposed in [40], though it does not explicitly account for the idle versus dynamic power characteristics of network devices. In our work, we reiterate power measurements and present a recent and more granular picture of the power variation of routers with respect to traffic properties and then extend the analysis to the flow level. Furthermore, ongoing efforts in IETF to define green metrics for networking [9, 22] only provide a high-level definition of metrics. Our work takes the definitions one step forward, refines the metrics, and shows their accuracy, scale and realistic usage for carbon tracing. This work aims to fill the gap for the flow-level energy and carbon metrics.

11 Conclusion

To leverage the benefits of carbon-aware networking requires carbon accounting at the flow level. Power models at the router level are not enough. This paper introduced a carbon emissions model for flows based on traffic statistics and network properties, derived from extensive power measurements on monolithic routers. Our regression model has an R^2 score $>96\%$ across all switch types and traces and an absolute mean percentage error of only 3–7% tested on a various Google services flows. The proposed model requires no hardware changes and can be deployed using existing monitoring tools and counters. Combined with server-generated carbon emissions, our model provides more holistic carbon footprint of networked applications.

Acknowledgments

This work was partly funded by NSF CBET-UKRI EPSRC TECAN (EP/X040828/1) and by the John Fell Oxford University Press Research Fund. We thank Tom King for his assistance with the configuration and maintenance of the experimental setup, as well as Itzik Kiselevsky, Golan Schzukin, Eve Schooler, and Robert Soulé for their constructive feedback.

For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission.

References

- [1] 2019. The CAIDA UCSD Anonymized Internet Traces - 2019. https://www.caida.org/catalog/datasets/passive_dataset [Online, accessed May 13, 2025].

- [2] 2021. Ethernet switching solutions at Broadcom. <https://www.broadcom.com/blog/ethernet-switching-solutions-at-broadcom> [Online, accessed July 17, 2025].
- [3] Bernardetta Addis, Antonio Capone, Giuliana Carello, Luca G Gianoli, and Brunilde Sansò. 2013. Energy Management through Optimized Routing and Device Powering for Greener Communication Networks. *IEEE/ACM Transactions on Networking* 22, 1 (2013), 313–325.
- [4] Akamai. 2023. 2023 ESG Impact Report. <https://www.akamai.com/site/en/documents/akamai/2024/akamai-2023-esg-impact-report.pdf> accessed 31 Jan 2025.
- [5] Baris Aksanli, Tajana Simunic Rosing, and Inder Monga. 2012. Benefits of Green Energy and Proportionality in High Speed Wide Area Networks Connecting Data Centers. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 175–180.
- [6] Vaastav Anand, Zhiqiang Xie, Matheus Stolet, Roberta De Viti, Thomas Davidson, Reyhaneh Karimipour, Safya Alzayat, and Jonathan Mace. 2023. The odd one out: Energy is not like other metrics. *ACM SIGENERGY Energy Informatics Review* 3, 3 (2023), 71–77.
- [7] AT&T. 2023. AT&T 2023 Sustainability Summary. <https://sustainability.att.com/ViewFile?fileGuid=032b23d3-77c3-4500-ae5a-84f74e646ca3> accessed 31 Jan 2025.
- [8] Aruna Prem Bianzino, Claude Chaudet, Dario Rossi, and Jean-Louis Rougier. 2010. A Survey of Green Networking Research. *IEEE Communications Surveys & Tutorials* 14, 1 (2010), 3–20.
- [9] Dean Bogdanović and Tony Li. 2024. Energy Metrics For Data Network. <https://www.ietf.org/archive/id/draft-bogdanovic-green-energy-metrics-00.txt> [Online, accessed April 20, 2025].
- [10] Roozbeh Bostandoost, Walid A Hanafy, Adam Lechowicz, Noman Bashir, Prashant Shenoy, and Mohammad Hajiesmaili. 2024. Data-driven Algorithm Selection for Carbon-Aware Scheduling. *ACM SIGENERGY Energy Informatics Review* 4, 5 (2024), 148–153.
- [11] Timm Böttger, Felix Cuadrado, Gareth Tyson, Ignacio Castro, and Steve Uhlig. 2018. Open connect everywhere: A glimpse at the Internet ecosystem through the lens of the Netflix CDN. *ACM SIGCOMM Computer Communication Review* 48, 1 (2018), 28–34.
- [12] Alasdair Bruce, Lyndon Ruff, James Kelloway, Fraser MacMillan, and Alex Rogers. 2020. Carbon Intensity Methodology. <https://www.carbonintensity.org.uk/> [Online, accessed May 13, 2025].
- [13] BT. 2024. Sustainable. <https://www.bt.com/about/digital-impact-and-sustainability/tackling-climate-change> accessed 31 Jan 2025.
- [14] Chen Chen, David Barrera, and Adrian Perrig. 2016. Modeling Data-Plane Power Consumption of Future Internet Architectures. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 149–158.
- [15] Hao Ran Chi, Daniel Corujo, Ayman Radwan, and Rui L. Aguiar. 2024. Metric Impact Towards Carbon-Aware Multi-domain Network Orchestration. In *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*. 110–115. doi:10.1109/GLOBECOM52923.2024.10901618
- [16] Luca Chiaraviglio, Antonio Cianfrani, Marco Listanti, Luigi Mignano, and Marco Polverini. 2015. Implementing Energy-Aware Algorithms in Backbone Networks: a Transient Analysis. In *2015 IEEE International Conference on Communications (ICC)*. IEEE, 142–148.
- [17] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Marazza, and Enrico Vittorini. 2010. An Energy Saving Routing Algorithm for a Green OSPF protocol. In *2010 INFOCOM IEEE Conference on Computer Communications Workshops*. IEEE, 1–5.
- [18] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, and Marco Polverini. 2011. An OSPF Enhancement for Energy Saving in IP Networks. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 325–330.
- [19] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Polverini, and Athanasios V Vasilakos. 2012. An OSPF-Integrated Routing Strategy for QoS-Aware Energy Saving in IP Backbone Networks. *IEEE Transactions on Network and Service Management* 9, 3 (2012), 254–267.
- [20] Quentin Clark, Fatih Acun, Ioannis C Paschalidis, and Ayse Coskun. 2024. Learning a Data Center Model for Efficient Demand Response. *ACM SIGENERGY Energy Informatics Review* 4, 5 (2024), 98–105.
- [21] Alexander Clemm, Dirk Kutscher, Michael Welzl, Cedric Westphal, Noa Zilberman, and Simone Ferlin-Reiter. 2025. Greening Networking: Toward a Net Zero Internet (Dagstuhl Seminar 24402). *Dagstuhl Reports* 14, 9 (2025), 167–192.
- [22] Alexander Clemm, Carlos Pignataro, Eve Schooler, Laurent Ciavaglia, Ali Rezaki, Greg Mirsky, and Jeff Tantsura. 2024. Green Networking Metrics for Environmentally Sustainable Networking. <https://datatracker.ietf.org/doc/draft-cx-green-green-metrics/> [Online, accessed April 20, 2025].
- [23] CloudFlare. 2025. Traffic - Cloudflare Radar. <https://radar.cloudflare.com/traffic/gb> [Online, accessed May 13, 2025].
- [24] Maurizio D'Arienzo and Simon Pietro Romano. 2016. GOSPF: An Energy Efficient Implementation of the OSPF Routing Protocol. *Journal of Network and Computer Applications* 75 (2016), 110–127.

- [25] 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. doi:10.1109/NoF62948.2024.10741444
- [26] Tamar Eilam, Pedro Bello-Maldonado, Bishwaranjan Bhattacharjee, Carlos Costa, Eun Kyung Lee, and Asser Tantawi. 2023. Towards a methodology and framework for AI sustainability metrics. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems*. 1–7.
- [27] Sawsan El-Zahr, Paul Gunning, and Noa Zilberman. 2023. Exploring the benefits of carbon-aware routing. *Proceedings of the ACM on Networking* 1, CoNEXT3 (2023), 1–24.
- [28] Sawsan El-Zahr, William Nathan, and Noa Zilberman. 2025. Carbon-Intelligent Content Scheduling in CDNs. In *Proceedings of the 2025 Applied Networking Research Workshop (Madrid, Spain) (ANRW '25)*. Association for Computing Machinery, New York, NY, USA, 1–8. doi:10.1145/3744200.3744774
- [29] Sawsan El Zahr and Noa Zilberman. 2025. Measurements to Emissions. <https://zenodo.org/records/17345074> [Online, accessed Oct 13, 2025].
- [30] Sawsan El Zahr and Noa Zilberman. 2025. Measurements2Emissions. <https://github.com/ox-computing/Measurements2Emissions> [Online, accessed Oct 13, 2025].
- [31] Anshul Gandhi, Dongyoon Lee, Zhenhua Liu, Shuai Mu, Erez Zadok, Kanad Ghose, Kartik Gopalan, Yu David Liu, Syed Rafiul Hussain, and Patrick Mcdaniel. 2023. Metrics for sustainability in data centers. *ACM SIGENERGY Energy Informatics Review* 3, 3 (2023), 40–46.
- [32] Erol Gelenbe and Christina Morfopoulou. 2011. A Framework for Energy-Aware Routing in Packet Networks. *Comput. J.* 54, 6 (2011), 850–859.
- [33] Google. 2024. Environmental Report 2024. <https://www.gstatic.com/gumdrop/sustainability/google-2024-environmental-report.pdf> accessed 31 Jan 2025.
- [34] P4.org Applications Working Group. 2020. In-band Network Telemetry (INT) Dataplane Specification. https://p4.org/p4-spec/docs/INT_v2_1.pdf accessed 15 May 2025.
- [35] Jiexiong Guan, Zhenqing Hu, Christos D. Antonopoulos, Nikolaos Bellas, et al. 2025. TMMModel: Modeling Texture Memory and Mobile GPU Performance to Accelerate DNN Computations. In *Proceedings of the ACM International Conference on Supercomputing*.
- [36] Yibo Guo, Amanda Tomlinson, Runlong Su, and George Porter. 2025. The Effect of the Network in Cutting Carbon for Geo-shifted Workloads. *arXiv preprint arXiv:2504.14022* (2025).
- [37] Leo Han, Jash Kakadia, Benjamin C Lee, and Udit Gupta. 2024. Towards game-theoretic approaches to attributing carbon in cloud data centers. In *Proceedings of the 2024 HotCarbon Workshop*. ACM.
- [38] Walid A Hanafy, Roozbeh Bostandoost, Noman Bashir, David Irwin, Mohammad Hajiesmaili, and Prashant Shenoy. 2024. The war of the efficiencies: Understanding the tension between carbon and energy optimization. *ACM SIGENERGY Energy Informatics Review* 4, 3 (2024), 87–93.
- [39] Sunpyo Hong and Hyesoon Kim. 2010. An integrated GPU power and performance model. In *Proceedings of the 37th annual international symposium on Computer architecture*. 280–289.
- [40] Kiyo Ishii, Junya Kurumida, Ken-ichi Sato, Tomohiro Kudoh, and Shu Namiki. 2015. Unifying top-down and bottom-up approaches to evaluate network energy consumption. *Journal of Lightwave Technology* 33, 21 (2015), 4395–4405.
- [41] Mark Jackson. 2023. Broadband ISP BT Sees Record UK Internet Traffic at 30.1Tbps. <https://www.ispreview.co.uk/index.php/2023/12/broadband-isp-bt-sees-record-uk-internet-traffic-of-30-1tbps.html> [Online, accessed May 14, 2025].
- [42] 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 on Internet Measurement Conference (IMC '25)*. Association for Computing Machinery, New York, NY, USA.
- [43] Romain Jacob and Laurent Vanbever. 2022. The Internet of tomorrow must sleep more and grow old. In *1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon)*.
- [44] Seng-Kyoun Jo, Lin Wang, Max Mühlhäuser, Young-Min Kim, and Jussi Kangasharju. 2017. Green Routing using Renewable Energy for IP Networks. In *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. IEEE, 1–6.
- [45] Hyeonwook Kim, Sydney Young, Xuesi Chen, Udit Gupta, and Josiah Hester. 2025. Slower is Greener: Acceptance of Eco-feedback Interventions on Carbon Heavy Internet Services. *ACM Journal on Computing and Sustainable Societies* 3, 2 (2025), 1–21.
- [46] Steven Lanzisera, Bruce Nordman, and Richard E Brown. 2012. Data Network Equipment Energy Use and Savings Potential in Buildings. *Energy Efficiency* 5 (2012), 149–162.
- [47] Electricity Maps. [n. d.]. Live 24/7 co2 emissions of electricity consumption. <https://app.electricitymaps.com/map> [Online, accessed May 13, 2025].

- [48] Julien Mineraud, Liang Wang, Sasitharan Balasubramaniam, and Jussi Kangasharju. 2016. Hybrid Renewable Energy Routing for ISP Networks. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.
- [49] national grid. 2023. What are scope 1, 2 and 3 carbon emissions? <https://www.nationalgrid.com/stories/energy-explained/what-are-scope-1-2-3-carbon-emissions#:~:text=Definitions%20of%20scope%201%2C%202,owned%20or%20controlled%20by%20it>.
- [50] Netflix. 2021. The True Climate Impact of Streaming. <https://about.netflix.com/en/news/the-true-climate-impact-of-streaming?> [Online, accessed May 13, 2025].
- [51] Netflix. 2023. Environmental Social Governance Report 2023. https://s22.q4cdn.com/959853165/files/doc_downloads/2024/6/2023-Netflix-Environmental-Social-Governance-Report.pdf accessed 31 Jan 2025.
- [52] Netflx. 2025. Internet connection speed recommendations. <https://help.netflix.com/en/node/306> [Online, accessed May 13, 2025].
- [53] R. Pant, T. Russell, C. Zorn, E. Oughton, and J.W. Hall. 2020. *Resilience Study Research for NIC – Systems Analysis of Interdependent Network Vulnerabilities*. Technical Report. Environmental Change Institute, Oxford University, UK.
- [54] Feitong Qiao, Yiming Fang, and Asaf Cidon. 2024. Energy-Aware Process Scheduling in Linux. *ACM SIGENERGY Energy Informatics Review* 4, 5 (2024), 91–97.
- [55] Xiaofeng Ren, Yiqing Wang, Hailong Gao, Xuequan Xiao, Yushan Zhou, and Hao Zhou. 2023. Tracing Carbon Responsibility Based on Carbon Emission Flow Tracking Theory. In *2023 6th International Conference on Energy, Electrical and Power Engineering (CEEPE)*. 1534–1538. doi:10.1109/CEEPE58418.2023.10166302
- [56] Shahbaz Rezaei and Xin Liu. 2018. How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets. *arXiv preprint arXiv:1812.09761* (2018).
- [57] Sergio Ricciardi, Davide Careglio, Francesco Palmieri, Ugo Fiore, Germán Santos-Boada, and Josep Solé-Pareta. 2011. Energy-Aware RWA for WDM Networks with Dual Power Sources. In *2011 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.
- [58] Ana Carolina Riekstin, Bruno Bastos Rodrigues, Kim Khoa Nguyen, Tereza Cristina Melo de Brito Carvalho, Catalin Meirosu, Burkhard Stiller, and Mohamed Cheriet. 2017. A survey on metrics and measurement tools for sustainable distributed cloud networks. *IEEE Communications Surveys & Tutorials* 20, 2 (2017), 1244–1270.
- [59] Sandvine. 2023. The Global Internet Phenomena Report January 2023. https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2023/reports/Sandvine%20GIPR%202023.pdf accessed 31 Jan 2025.
- [60] Andreas Schmidt, Gregory Stock, Robin Ohs, Luis Gerhorst, Benedict Herzog, and Timo Hönig. 2024. carbond: An operating-system daemon for carbon awareness. *ACM SIGENERGY Energy Informatics Review* 4, 3 (2024), 52–57.
- [61] 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*. 144–158.
- [62] Yuta Tokusashi, Huynh Tu Dang, Fernando Pedone, Robert Soulé, and Noa Zilberman. 2019. The case for in-network computing on demand. In *Proceedings of the Fourteenth EuroSys Conference 2019*. 1–16.
- [63] Karel van der Veldt, Cees de Laat, Inder Monga, Jon Dugan, and Paola Grosso. 2014. Carbon-Aware Path Provisioning for NRENs. In *International Green Computing Conference*. IEEE, 1–7.
- [64] Nedeljko Vasić, Prateek Bhurat, Dejan Novaković, Marco Canini, Satyam Shekhar, and Dejan Kostić. 2011. Identifying and Using Energy-Critical Paths. In *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies*. 1–12.
- [65] Nedeljko Vasić and Dejan Kostić. 2010. Energy-Aware Traffic Engineering. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*. 169–178.
- [66] Arun Vishwanath, Kerry Hinton, Robert WA Ayre, and Rodney S Tucker. 2014. Modeling Energy Consumption in High-Capacity Routers and Switches. *IEEE Journal on Selected Areas in Communications* 32, 8 (2014), 1524–1532.
- [67] Yuan Yang, Dan Wang, Dawei Pan, and Mingwei Xu. 2016. Wind Blows, Traffic Flows: Green Internet Routing under Renewable Energy. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.
- [68] Yuan Yang, Mingwei Xu, Dan Wang, and Suogang Li. 2015. A Hop-by-Hop Routing Mechanism for Green Internet. *IEEE Transactions on Parallel and Distributed Systems* 27, 1 (2015), 2–16.
- [69] Mingui Zhang, Cheng Yi, Bin Liu, and Beichuan Zhang. 2010. GreenTE: Power-Aware Traffic Engineering. In *The 18th IEEE International Conference on Network Protocols*. IEEE, 21–30.
- [70] Xinnian Zheng, Lizy K John, and Andreas Gerstlauer. 2017. LACross: Learning-based analytical cross-platform performance and power prediction. *International Journal of Parallel Programming* 45, 6 (2017), 1488–1514.
- [71] Noa Zilberman, Gabi Bracha, and Golan Schuzkin. 2019. Stardust: Divide and conquer in the data center network. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 141–160.
- [72] Noa Zilberman, Eve M Schooler, Uri Cummings, Rajit Manohar, Dawn Nafus, Robert Soulé, and Rick Taylor. 2022. Toward Carbon-Aware Networking. In *1st Workshop on Sustainable Computer Systems Design and Implementation*

(HotCarbon).

A Counters and Metrics

Table 8 clarifies the counters and metrics needed for every carbon emissions scope of flows.

Table 8. Counters and Metrics per Carbon type

Carbon Type	Counters					Metrics					Formula
	Router Bytes (RB)	Router Pkts (RP)	Router Nb Flows (RF)	Flow Bytes (FB)	Flow Pkts (FP)	Energy/B (α)	Energy/Pkt (β)	Idle Energy (IE)	Max Tput (Cmax)	Carbon Intensity (CI)	
Consequential Carbon				✓	✓	✓	✓			✓	$(FB * \alpha + FP * \beta) * CI$
Attributional (Nb Flows)			✓					✓		✓	$IE * (1/RF) * CI$
Attributional (Nb Flows + U)	✓		✓					✓	✓	✓	$IE * (1/RF) * (RB/Cmax) * CI$
Attributional (BitRate)	✓			✓				✓		✓	$IE * (FB/RB) * CI$
Attributional (BitRate + U)	✓			✓				✓	✓	✓	$IE * (FB/RB) * (RB/Cmax) * CI$
Attributional (PktRate)		✓			✓			✓		✓	$IE * (FP/RP) * CI$
Attributional (PktRate + U)	✓	✓			✓			✓	✓	✓	$IE * (FP/RP) * (RB/Cmax) * CI$

Received July 2025; revised September 2025; accepted October 2025