Not Just Fast, But Also Sustainable: Rethinking Network Routing

AMIT SAMANTA, The University of Utah, Salt Lake City, UT, United States

YANKAI JIANG, Northeastern University, Boston, MA, United States

RYAN STUTSMAN, The University of Utah, Salt Lake City, UT, United States

ROHAN BASU ROY, The University of Utah, Salt Lake City, UT, United States

Published: 15 December 2025

Citation in BibTeX format

MIDDLEWARE '25: 26th International Middleware Conference
December 15 - 19, 2025
TN, Nashville, USA

Conference Sponsors:
ACM

MAIoT '25: Proceedings of the Middleware for Autonomous AIoT Systems in the Computing Continuum (December 2025)
https://doi.org/10.1145/3774901.3778064
ISBN: 9798400723049

Not Just Fast, But Also Sustainable: Rethinking Network Routing

Amit Samanta
University of Utah
Salt Lake City, USA

Yankai Jiang
Northeastern University
Boston, USA

Ryan Stutsman
University of Utah
Salt Lake City, USA

Rohan Basu Roy
University of Utah
Salt Lake City, USA

Abstract

The carbon footprint of networking poses a significant environmental concern. In this paper, we show how transmission latency-focused traditional routing can be sub-optimal in terms of carbon footprint. Based on the dynamic factors that affect a network's carbon footprint and transmission latency, we design a carbon-aware routing solution. We evaluate our solution on real backbone Internet topologies to show that there is an opportunity to minimize the network's carbon emissions with only modest latency penalties.

CCS Concepts

• **Networks → Network protocols**.

Keywords

Sustainability, Routing Protocol, Carbon Emissions.

ACM Reference Format:
Amit Samanta, Yankai Jiang, Ryan Stutsman, and Rohan Basu Roy. 2025. Not Just Fast, But Also Sustainable: Rethinking Network Routing . In *Middleware for Autonomous AIoT Systems in the Computing Continuum (MAIoT'25), December 15–19, 2025, Nashville, TN, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3774901.3778064

1 Introduction

Why should we worry about networking's carbon footprint? The carbon footprint of computing systems is rapidly becoming a critical concern with the increase in AI/ML workloads, expanding data volumes, and the continuous development of advanced hardware and networking equipments. Recent studies suggest that network infrastructures alone consume approximately 1.5× the electricity used by data centers, contributing to the broader ICT sector's 2–3% share of global electricity usage, a figure projected to climb to as high as 8–21% by 2030 [9, 22, 36]. Despite this, much of the existing work on sustainability has focused on high-performance computing cloud resource management, and hardware optimizations [7, 14, 16, 21, 23, 25, 27, 35], often overlooking the rising impact of network traffic on overall carbon emissions.

How is carbon-aware routing different from traditional protocols? Conventional routing protocols (e.g., OSPF, BGP) perform latency optimization, overlooking how energy usage and carbon intensity fluctuate with interface type, traffic load, and location. Because carbon intensity shifts both spatially and temporally, a path that minimizes carbon footprint may differ significantly from one that minimizes latency. This gap reveals an opportunity to integrate carbon-awareness into routing, dynamically adapting to congestion and time-varying carbon data. By coupling conventional latency objectives with the demand for greener operation, carbon-aware routing provides a mechanism to substantially reduce emissions without sacrificing network performance much.

Contributions. In this paper, we make four key contributions:

- This work showcases the benefit of carbon-aware routing on large-scale real Internet backbone topologies.
- We present a design that jointly minimizes latency and carbon footprint, which can be implemented on top of the existing routing schemes.
- We evaluate our approach on real Internet topologies with live carbon intensity data, showing significant carbon savings with some latency penalty.
- We discuss future steps to improve the robustness of carbon-focused routing, envisioning this work as a foundation for future research.

2 Background and Motivation

We provide a brief background on carbon footprint and routing before motivating the potential of carbon-aware routing.

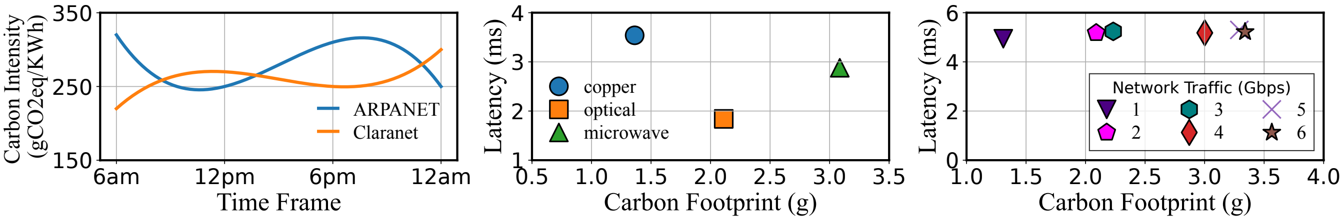
Carbon Footprint and Sustainability. Computing's energy consumption is increasing due to the rising demand for AI/ML workloads, expanding data traffic, and the production of advanced networking and computing hardware [34]. Energy production and hardware manufacturing emit greenhouse gases, thus contributing to computing's environmental sustainability concerns. Carbon footprint is the key sustainability metric; it includes the amount of CO_2 and the equivalent of other greenhouse gases emitted for operating (*operational carbon*) and manufacturing a system (*embodied carbon*) [13]. Operational emissions are the product of energy consumption and carbon intensity (CI; CO_2-equivalent greenhouse gas emitted per unit of energy production), while embodied emissions are one-time emissions from hardware manufacturing. In networking, operational emissions dominate due to high routing energy use [36].

Routing Mechanisms. Network routing mechanisms, such as distance-vector (e.g., RIP), link-state (e.g., OSPF), and path-vector (e.g., BGP), form the backbone of modern data transfer and are optimized for performance by minimizing transmission latency. However, these protocols largely overlook the growing carbon footprint of data transmission. There is an urgency of incorporating carbon-awareness into routing strategies as the Internet accounts for ≈3% of global emissions which is increasing at a rate of 5-7% [3, 6].

How to Perform Carbon-Aware Routing in Networks? Network carbon footprint can be minimized by leveraging the spatio-temporal variation of carbon intensity [28]. Power grids of different

MAIoT'25, Nashville, TN, USA

ACM ISBN 979-8-4007-2304-9/2025/12
https://doi.org/10.1145/3774901.3778064



(a) (b) (c)

Figure 1: Carbon intensity (a), type of interface (b), and router traffic load (c) affect carbon-aware routing.

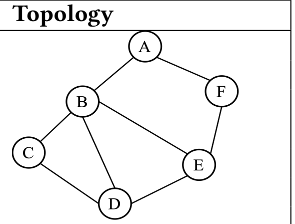
Node	Interface	Topology
A	Copper	
B	Microwave	
C	Optical	
D	Microwave	
E	Optical	
F	Copper	

Table 1: We present a representative topology. Carbon-optimal and latency-optimal paths differ and vary over time. At 12 AM, the latency-opt path (A->B->E->F) had a latency of 3.12ms and carbon of 5.14g, while the carbon-opt path (A->B->D->E->F) had 3.24ms latency and 4.17g carbon. At 3 PM, the latency-opt path shifted to (A->B->D->E->F) with 3.28ms latency and 6.52g carbon, and the carbon-opt path to (A->C->D->E->F) with 3.48ms latency and 5.20g carbon.

locations use different energy source mixes to produce energy. Energy generation using a large fraction of renewable sources results in low carbon intensity, while generation consuming emission-heavy sources like fossil fuel results in higher carbon intensity [10].

The energy consumed by transmission interfaces (e.g., copper, microwave, optical) varies. Transmission carbon footprint is the product of router and interface energy consumption and the grid's carbon intensity, with embodied carbon being negligible. The factors affecting the carbon footprint of transmission in a network (the carbon intensity of the router locations, and energy consumption based on load) are different from the factors affecting transmission latency in a network (depends on the number of hops, transmission interface, and congestion). This is why the carbon-optimal routing path is different than traditional latency-optimal routing paths. Table 1 establishes that.

Table 1 shows a subset of the original ARPANET network topology, based on its initial node locations across major research institutions in the United States. For these nodes, we consider the average daily variation in carbon intensity, the network load at each router location, and the predominant transmission interface used. At 12 AM, the latency optimal path is A->B->E->F, which is different from the carbon-optimal path A->B->D->E->F. The carbon-optimal path reduces carbon footprint by 18%, however, increasing latency by 11%. At a different time of the day, 3 PM, both the latency and carbon-optimal path changes. Latency-optimal paths change with traffic and congestion, while carbon-optimal paths shift with router carbon intensity and traffic-driven energy use. These dynamics highlight the opportunities for adaptive strategies to balance carbon footprint and latency.

Takeaway. Network carbon footprint is often overlooked, despite being a significant contributor to the ICT industry's emissions. Traditional latency-focused routing solutions are not carbon-efficient. Carbon- and latency-optimal routing paths are different and they vary dynamically over time due to changes in traffic patterns, congestion, and carbon intensity, highlighting the need for adaptive routing strategies.

Factors Impacting Carbon-Aware Routing. Carbon intensity is a key factor that impacts a network's carbon footprint As noted before carbon intensity changes temporally and also across locations, depending on the energy source mix used for power generation. Fig. 1(a) shows the carbon intensity variations of two networks, ARPANET and Claranet (details in Sec. 4). If a data packet can take a route among any of these two networks, at 6 AM it is beneficial to choose Arpanet, while at 6 PM it is beneficial to choose Claranet, based on the corresponding lower values of carbon intensity. This spatio-temporal variation of carbon intensity makes carbon-aware routing fundamentally different from traditional latency-focused routing, as carbon-aware routing should dynamically adapt to both time and location.

Next, Fig. 1(b) illustrates the trade-offs between carbon footprint and latency for copper, optical, and microwave interfaces for transmitting data at a fixed data transfer rate of 1 Gbps in the Claranet topology. Optical suits low latency and moderate carbon intensity, copper prioritizes sustainability over latency, and microwave fits long-range or limited-interface scenarios. Dynamic transmission interface selection also offers an opportunity for real-time optimization of performance and environmental impact.

Finally, Fig. 1(c) shows the carbon footprint and latency for transmitting data as traffic load at routers varies from 1 to 6 Gbps for the ARPANET topology. For this experiment, a hybrid interface combining optical, copper, and microwave technologies was used. The results demonstrate that both carbon footprint and latency increase with higher traffic levels. This is because as traffic increases, the energy consumption of routers and interfaces rises superlinearly, leading to greater carbon emissions. Higher traffic causes congestion, increasing latency from longer queuing and processing. This emphasizes the need for routing solutions that adapt to traffic loads to balance latency and carbon footprint.

Takeaway. Carbon intensity varies spatio-temporally, requiring adaptive strategies to reduce a network's carbon footprint. Transmission interfaces offer distinct trade-offs, enabling dynamic selection based on carbon and latency goals. Rising traffic amplifies congestion and emissions, necessitating carbon-aware routing to balance performance and sustainability.

3 Design: Carbon-Aware Routing Framework

We propose a carbon-aware routing framework that can work standalone or with protocols like BGP. Using a modified Dijkstra's algorithm, it dynamically selects paths and interfaces to minimize latency and carbon footprint based on real-time congestion, carbon intensity, and congestion.

3.1 Problem Formulation

Let $G = (V, E)$ represent the network graph, where each vertex $r \in V$ is a router and each edge $(u, v) \in E$ is a direct link between routers u and v. Each router r has an associated, time-varying carbon intensity $\mathrm{CI}_t(r)$ (experimental values obtained from Electricity Maps [19]) depending upon the real-time mix of energy sources used to produce electricity at the router's location. For each link (u, v), the interface type $I_{uv} \in$ {copper, microwave, optical fiber} affects latency and power consumption (which in turn affects the carbon footprint). Router congestion introduces additional latency, denoted by $\mathrm{L}_{\mathrm{cong}}(C_r)$, and scales the overall power usage through a function $f(C_r)$, where C_r is the load at router r. Given a path P between a source s and a destination d, the total routing cost is:

$$\mathrm{Cost}(P) = \sum_{(u,v)\in P} \Big(\lambda_p \cdot L_{uv} + (1-\lambda_p) \cdot C_{uv}\Big), \tag{1}$$

where $\lambda_p \in [0, 1]$ is a sensitivity factor that balances latency (L_{uv}) and carbon (C_{uv}). We try to minimize this cost (P) by dynamically selecting the optimal path and interface type for each link (u, v) to achieve an efficient trade-off between latency and carbon footprint.

Routing Cost Components Next, we define the components of P. Each link (u, v) is associated with a latency term:

$$L_{uv} = L_{\mathrm{base}}(I_{uv}) + L_{\mathrm{cong}}(C_r). \tag{2}$$

$L_{\mathrm{base}}(I_{uv})$ is the base latency of the selected interface (copper, microwave, or optical fiber), while $\mathrm{L}_{\mathrm{cong}}(C_r)$ reflects congestion-based delays that grow with router load. The carbon cost on link (u, v) is given by:

$$C_{uv} = \Big(P_{\mathrm{base}}(I_{uv}) + P_{\mathrm{dyn}}(I_{uv}) \cdot S_p \cdot f(C_r)\Big) \cdot \mathrm{CI}_t(r), \tag{3}$$

where $P_{\mathrm{base}}(I_{uv})$ is the base (or idle) power consumption required to keep the interface active, even if no data is being transmitted. $P_{\mathrm{dyn}}(I_{uv})$ is the dynamic power component, which scales with the amount of data being sent and represents the additional energy required per unit of data. S_p denotes the packet size, and $\mathrm{CI}_t(r)$ is the real-time carbon intensity at router r. $f(C_r)$ is a scaling function applied to the dynamic power usage, reflecting how higher load or congestion at router r increases total power requirements. Putting these terms together, the product $P_{\mathrm{dyn}}(I_{uv}) \cdot S_p \cdot f(C_r)$ captures the total additional power overhead incurred for forwarding a packet under current load conditions, and $P_{\mathrm{base}}(I_{uv})$ provides a baseline power cost for maintaining the interface. Multiplying by $\mathrm{CI}_t(r)$ converts the resulting power cost into the carbon footprint for link (u, v).

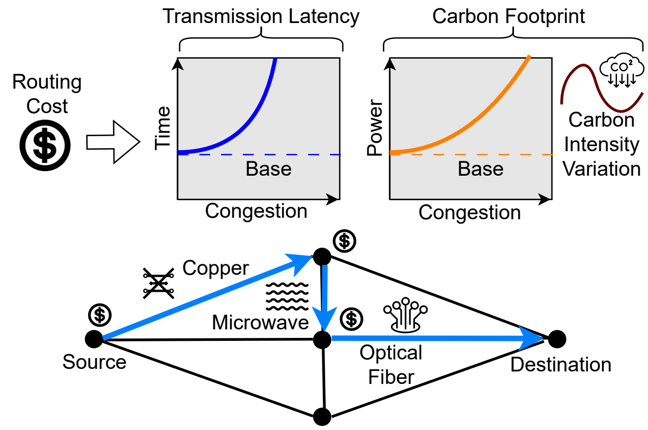


Figure 2: Carbon-aware routing framework dynamically selects optimal paths and interfaces based on real-time metrics such as carbon intensity variations, congestion levels, latency, and power consumption.

3.2 Carbon-Aware Path Optimization

Our approach uses a modified Dijkstra's algorithm to perform a network's routing decision. It incorporates real-time carbon intensity variations, congestion, and interface-specific latencies. Border Gateway Protocol (BGP) remains the de facto standard for inter-domain routing. However, it primarily focuses on policy enforcement, and does not perform optimization based on dynamic metrics such as real-time carbon intensity. By contrast, Dijkstra's algorithm naturally represents the network as a weighted graph, enabling a more granular and flexible approach to path selection. This method can be layered on top of existing BGP deployments or used standalone, allowing domains to internally optimize for sustainability while maintaining global reachability.

Dynamic Workflow Optimization. Classical Dijkstra's algorithm computes the shortest path by prioritizing nodes with the lowest cumulative cost. We extend it by incorporating real-time metrics, including fluctuating load, carbon intensity, and interface-specific latency and power consumption, into the link cost. Routing iteratively selects the next router, considering its location, and the interface type (I_{uv}) to minimize cumulative routing cost from source to destination.

To initialize the system, we first construct the network graph $G = (V, E)$ and associate each link (u, v) with its baseline latency and power attributes. We then perform periodic updates to both $\mathrm{CI}_t(r)$ (reflecting the location-specific and temporal carbon intensity), router load C_r (influencing congestion latency $\mathrm{L}_{\mathrm{cong}}(C_r)$ and the scaling function $f(C_r)$). Whenever these parameters change significantly, the cost-aware routing logic recalculates the path from a source s to a destination d. Once a cost-optimal path is selected, packets are forwarded along the links using the chosen interface types, thereby reflecting the updated latency and carbon considerations in near real-time.

Interface Selection Criteria. Each interface type (copper, microwave, and optical) exhibits distinct power requirements and base latencies. The selection of I_{uv} relies on comparing:

$$\mathrm{Cost}(I_{uv}) = \lambda_p \cdot \mathrm{L}_{\mathrm{base}}(I_{uv}) + (1-\lambda_p) \cdot P_{uv} \cdot \mathrm{CI}_t(r) \tag{4}$$

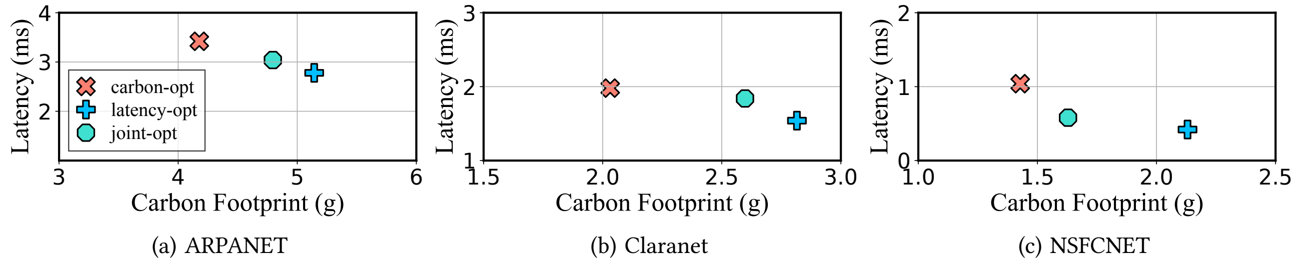


Figure 3: There exists an opportunity to balance sustainability and performance through carbon-aware routing.

where P_{uv} corresponds to the power demands of the respective interface. By evaluating Eq. 4 in real time for each possible interface, the system selects the most suitable link-layer technology on a per-hop basis to reflect both performance requirements and carbon-conscious policies.

Scalability and Real-Time Adaptation. The system updates link weights incrementally with new $\text{CI}_t(r)$ and C_r values, avoiding full recomputation. In large-scale deployments, path computation can be parallelized or distributed, and in SDN settings, the optimizer can update flow rules via controller APIs. This ensures robust real-time adaptation for optimizing latency and carbon footprint.

3.3 Putting It All Together

Our carbon-aware routing normalizes latency and carbon costs using min-max normalization for comparability. It dynamically updates forwarding tables or SDN flow rules to minimize combined costs. Fig. 2 illustrates path and interface selection (e.g., Copper, microwave, optical) based on real-time updates to carbon intensity, congestion, and latency. Routing costs vary with router congestion, and the system continuously monitors the network state, recalibrates link weights, and chooses carbon-efficient paths, ensuring scalability while balancing latency and emissions.

4 Evaluation

Here we discuss our evaluation methodology and the results.

4.1 Evaluation Methodology

Evaluation with Real-World Network Topologies and Carbon Intensity. We use the Internet Topology Zoo [4] to analyze the carbon intensity and routing latency of different diverse topologies: ARPANET (an early backbone network in the USA), Claranet (a European service provider network), and NSFCNET (a research network topology in China). While these are not current global backbone networks, they showcase diverse geographic scopes, infrastructure types, and connectivity characteristics, making them representative for evaluating carbon-aware routing across foundational, dense regional, and large-scale, energy-intensive infrastructures. We design a topology-driven event simulator for evaluating network routing. We use Electricity Maps [19], a widely used tool to determine real-time energy grid-related metrics, to determine the carbon intensity of the router locations in the network topologies. We consider three types of interfaces: copper, optical fiber, and microwave. The power consumption rate of the interfaces is set to their representative values: copper's rate is 2-5W [1], optical fiber's rate is 10mW [5], and microwave's rate is 358W [2]. Along with the energy consumption of interfaces, we consider router's energy consumption for calculating carbon footprint.

Evaluated Schemes. We evaluate **Carbon-optimal** (carbon-opt), **Latency-optimal** (latency-opt), and **Joint-optimal** (joint-opt). Carbon-opt dynamically selects network paths that minimize only carbon emissions associated with transmission (λ_p in Eq. 1 is set to 0). Latency-opt prioritizes network paths that minimize only time taken for data packets to travel from source to destination (λ_p is set to 1). Joint-opt achieves a balance between minimizing latency and carbon emissions during data transmission (λ_p is set to 0.5).

Metrics. We evaluate carbon footprint and routing latency. Carbon footprint is calculated by multiplying router and interface energy consumption by the carbon intensity of the corresponding energy grid. Latency measures the time for data packets to travel from source to destination.

4.2 Results

In Fig. 3, we compare the three evaluated schemes—*carbon-opt*, *latency-opt*, and *joint-opt*—across the ARPANET, Claranet, and NSFCNET topologies. For each topology, we run a one-day experiment using hourly variations of real-world carbon intensity (CI) data from the nodes' respective geographic locations, while also varying congestion and load conditions. At each time instant, our optimizer selects both the route and the interface type (copper, microwave, or optical fiber) to balance latency and power consumption based on carbon intensity variations and congestion levels. In the ARPANET example (Fig. 3(a)), the *carbon-opt* strategy achieves the lowest average carbon footprint (4.17g) at the cost of higher latency (3.42ms) when transmitting at an average rate of 1Gbps. Conversely, the *latency-opt* strategy yields the smallest latency (2.78ms) but incurs a higher carbon footprint (5.14g). The *joint-opt* method finds a middle ground, reducing carbon footprint to 4.79g while maintaining latency near 3.04ms. These results underscore the trade-off between environmental impact and performance. Joint-opt balances significant carbon savings with minimal latency increase.

Figs. 3(b) and 3(c) show similar trends for Claranet and NSFCNET, with varying carbon and latency savings. Claranet's lower carbon intensity reduces the gap between joint- and latency-opt solutions, while NSFCNET's higher intensity allows greater carbon savings, with joint-opt nearing carbon-opt while maintaining moderate latency benefits.

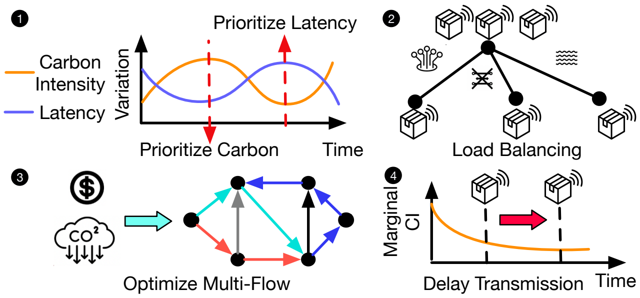


Figure 4: Future directions in carbon-aware routing.

Takeaway. The carbon-opt approach cuts emissions at the expense of higher latency, while latency-opt achieves lower delays and higher carbon footprint. By offering a balanced trade-off, the joint-opt strategy demonstrates how carbon-aware routing can significantly curb emissions without compromising performance much.

5 Discussions and Future Directions

Fig. 4 summarizes the potential future research directions.

1. Dynamic Prioritization of Latency and Carbon. λ_p is introduced as a fixed parameter that controls the trade-off between optimizing for latency and carbon footprint. Allowing λ_p to dynamically adapt in the framework can offer greater benefits. For example, prioritizing carbon footprint optimization can yield more significant benefits compared to focusing on latency optimization, if the carbon intensity variation is larger than the latency variation of interfaces.

2. Load Balancing Based on Interface. Different interfaces respond differently to congestion. Optical links can handle more traffic with a smaller energy increase, whereas copper might exhibit a steep power jump under similar loads. Balancing traffic to match interface energy minimizes carbon.

3. Adaptive Traffic Engineering Across Multiple Flows. The framework can be extended to consider the interactions among a broader set of simultaneous data transfers. This approach optimizes carbon and latency across the network, ensuring balanced and efficient traffic distribution while considering flow interdependencies.

4. Delay Transmission. Delaying packet transmission can provide temporal benefits by waiting for periods of lower carbon intensity to reduce the carbon footprint. This approach leverages the frequent fluctuations in marginal carbon intensity across the energy grid, allowing the network to schedule transmissions during more sustainable periods.

6 Related Works and Conclusion

Environmental sustainability is a growing concern in the computer systems community, which has been mostly focused on datacenter scheduling and hardware resource optimization to reduce carbon emissions, while sustainable routing in large-scale networks remains largely underexplored [7, 8, 11, 13–15, 17, 18, 25, 33]. Sawsan *et al.* [12] introduced CATE (carbon-aware traffic engineering) to reduce emissions via dynamic routing. Singh *et al.* [24] explored carbon-aware routing in software-defined, inter-datacenter networks. Tabaeiaghdaei *et al.* [30] proposed CIRo for path-aware networks, forecasting and disseminating carbon intensity for inter-domain routes. Van *et al.* [32] focused on carbon-aware provisioning for NRENs by integrating real-time grid carbon data with path selection. While these works prioritize emissions reduction, some neglect latency and rely on simplified models or specialized environments [12, 20, 24, 26, 29–32], limiting generality across the broader Internet.

In contrast, this work presents a general opportunity to perform carbon-aware routing on the Internet or large-scale communication networks without severely compromising performance. Our results show that focusing strictly on carbon reduction can cut emissions compared to latency-centric strategies but incur a higher latency overhead, while a joint carbon-latency optimization approach offers both moderate carbon savings and performance penalty. These findings suggest the potential for balancing carbon awareness and latency for sustainable network routing.

Acknowledgments: We thank the anonymous reviewers for their valuable feedback. This work is supported by the University of Utah's Kahlert School of Computing, John and Marcia Price College of Engineering, and Scientific Computing & Imaging Institute.

References

[1] [n.d.]. 10G Network Upgrades: 3 Reasons To Choose Copper. https://www.precisionot.com/10g-network-upgrades-3-reasons-choose-copper/.
[2] [n.d.]. Coaxial Cable Power Handling. https://www.microwavejournal.com/articles/9174-coaxial-cable-power-handling.
[3] [n.d.]. Do I emit CO2 when I surf the internet? — energuide.be. https://www.energuide.be/en/questions-answers/do-i-emit-co2-when-i-surf-the-internet/69/.
[4] [n.d.]. The Internet Topology Zoo. https://topology-zoo.org/dataset.html.
[5] [n.d.]. Optical Power. https://www.thefoa.org/tech/ref/testing/test/power.html.
[6] [n.d.]. The Carbon Footprint of the Internet: How Your Data Usage Emits CO2 - Shift Blog | Browser Tips, App Integrations, and Productivity — shift.com. https://shift.com/blog/news/the-carbon-footprint-of-the-internet/.
[7] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon explorer: A holistic framework for designing carbon aware datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 118–132.
[8] Thomas Anderson, Adam Belay, Mosharaf Chowdhury, Asaf Cidon, and Irene Zhang. 2023. Treehouse: A case for carbon-aware datacenter software. *ACM SIGENERGY Energy Informatics Review* 3, 3 (2023), 64–70.
[9] Anders SG Andrae and Tomas Edler. 2015. On global electricity usage of communication technology: trends to 2030. *Challenges* 6, 1 (2015), 117–157.
[10] Sally M Benson and Franklin M Orr. 2008. Carbon dioxide capture and storage. *MRS bulletin* 33, 4 (2008), 303–305.
[11] Lieven Eeckhout. 2024. FOCAL: A First-Order Carbon Model to Assess Processor Sustainability. In *ACM ASPLOS*. 401–415.
[12] Sawsan El-Zahr, Paul Gunning, and Noa Zilberman. 2023. Exploring the Benefits of Carbon-Aware Routing. *ACM CoNEXT* 1 (2023), 1–24.
[13] Udit Gupta, Mariam Elgamal, Gage Hills, Gu-Yeon Wei, Hsien-Hsin S Lee, David Brooks, and Carole-Jean Wu. 2022. ACT: Designing sustainable computer systems with an architectural carbon modeling tool. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 784–799.
[14] Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. 2021. Chasing carbon: The elusive environmental footprint of computing. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 854–867.
[15] Walid A Hanafy, Qianlin Liang, Noman Bashir, Abel Souza, David Irwin, and Prashant Shenoy. 2024. Going Green for Less Green: Optimizing the Cost of Reducing Cloud Carbon Emissions. In *ACM ASPLOS*. 479–496.
[16] Paras Jain, Sam Kumar, Sarah Wooders, Shishir G Patil, Joseph E Gonzalez, and Ion Stoica. 2023. Skyplane: Optimizing Transfer Cost and Throughput Using {Cloud-Aware} Overlays. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1375–1389.
[17] Yankai Jiang, Rohan Basu Roy, Baolin Li, and Devesh Tiwari. 2024. EcoLife: Carbon-Aware Serverless Function Scheduling for Sustainable Computing. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–15.
[18] Baolin Li, Rohan Basu Roy, Daniel Wang, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. 2023. Toward sustainable hpc: Carbon footprint estimation and environmental implications of hpc systems. In *ACM/IEEE SC*. 1–15.
[19] Electricity Maps. 2025. *Electricity Maps Live 24/7*. https://app.electricitymaps.com/map
[20] Chung-Lam Ng, Hang Wu, Ming Li, Ray Y Zhong, Xinye Qu, and George Q Huang. 2024. A carbon-aware routing protocol for optimizing carbon emissions in modular construction

logistics. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2852–2856.
[21] Konstantinos Partsafas. 2023. The advantages of cloud computing over traditional IT infrastructure and services. (2023).
[22] Anders SG Andrae. 2020. New perspectives on internet electricity use in 2030. *Engineering and Applied Science Letter* 3, 2 (2020), 19–31.
[23] Prateek Sharma. 2022. Challenges and opportunities in sustainable serverless computing.
[24] Deepshikha Singh and Gitanjali Chandwani. 2015. Carbon-aware routing in software defined inter data center network. In *IEEE ANTS*. 1–6.
[25] Abel Souza, Noman Bashir, Jorge Murillo, Walid Hanafy, Qianlin Liang, David Irwin, and Prashant Shenoy. 2023. Ecovisor: A virtual energy system for carbon-efficient applications. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 252–265.
[26] Abel Souza, Shruti Jasoria, Basundhara Chakrabarty, Alexander Bridgwater, Axel Lundberg, Filip Skogh, Ahmed Ali-Eldin, David Irwin, and Prashant Shenoy. 2023. CASPER: Carbon-Aware Scheduling and Provisioning for Distributed Web Services. In *Proceedings of the 14th International Green and Sustainable Computing Conference*. 67–73.
[27] Soumya Sudhakar, Vivienne Sze, and Sertac Karaman. 2022. Data centers on wheels: emissions from computing onboard autonomous vehicles. *IEEE Micro* 43, 1 (2022), 29–39.
[28] Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. 2024. On the Limitations of Carbon-Aware Temporal and Spatial Workload Shifting in the Cloud. In *ACM EuroSys*. 924–941.
[29] Edwin Sutherland and Iain Phillips. 2024. Low-Carb BGP: A Carbon-Aware Inter-Domain Routing Extension to BGP. In *Proceedings of the 2024 Applied Networking Research Workshop*. 53–59.
[30] Seyedali Tabaeiaghdaei, Simon Scherrer, Jonghoon Kwon, and Adrian Perrig. 2022. Carbon-Intelligent Global Routing in Path-Aware Networks. *arXiv preprint arXiv:2211.00347* (2022).
[31] 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.
[32] Karel van der Veldt, Cees de Laat, Inder Monga, Jon Dugan, and Paola Grosso. 2014. Carbon-aware path provisioning for NRENs. In *IEEE International Green Computing Conference*. 1–7.
[33] Jaylen Wang, Daniel S Berger, Fiodar Kazhamiaka, Celine Irvene, Chaojie Zhang, Esha Choukse, Kali Frost, Rodrigo Fonseca, Brijesh Warrier, Chetan Bansal, et al. 2024. Designing cloud servers for lower carbon. In *ACM/IEEE ISCA*. 452–470.
[34] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems* 4 (2022), 795–813.
[35] Zongheng Yang, Zhanghao Wu, Michael Luo, Wei-Lin Chiang, Romil Bhardwaj, Woosuk Kwon, Siyuan Zhuang, Frank Sifei Luan, Gautam Mittal, Scott Shenker, et al. 2023. {SkyPilot}: An Intercloud Broker for Sky Computing. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 437–455.
[36] Noa Zilberman, Eve M Schooler, Uri Cummings, Rajit Manohar, Dawn Nafus, Robert Soulé, and Rick Taylor. 2023. Toward carbon-aware networking. *ACM SIGENERGY Energy Informatics Review* 3, 3 (2023), 15–20.