Designing Efficient and Equitable Networked Systems for Mobile Users in Emerging Regions

by

Rohail Asim

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

August, 2025

Professor Yasir Zaki

Professor Lakshmi Subramanian

Dedication

To my family, for their uwavering love and support

Acknowledgements

I would like to express my profound gratitude to my advisors, Professor Yasir Zaki and Professor Lakshmi Subramanian. Their mentorship, guidance, care, and continuous support throughout my PhD journey have been invaluable. Coming from an underdeveloped region, I was immediately drawn to Yasir's work on improving web accessibility for emerging regions through research that does not simply exist behind a screen, but has the potential to make a tangible impact in the real world. I am grateful to have had the opportunity to carry forward this philosophy in my research. I would also like to thank my committee members: Talal Rahwan, Matteo Varvello, and Anirudh Sivaraman for their valuable feedback and encouragement.

I am immensely grateful to my mentors, collaborators, colleagues, and friends: Professor Fareed, for the opportunity to work at the TPI Lab as an undergraduate research assistant, an invaluable experience that prepared me for my PhD journey; and all of my friends and collaborators at NYU and NYU Abu Dhabi with whom I worked on many exciting projects – Muhammad Khan, Ayush Pandey, Moumena Chaqfeh, Hazem Ibrahim, and Ankit Bhardwaj. I would also like to thank everyone working behind the scenes at NYU and NYUAD, especially Karl, Ross, and Santiago, for ensuring a prompt resolution to any administrative issues.

Finally, I owe everything to my family. To my parents, for their unwavering love and for supporting my dreams no matter how far they took me. And to my wife, Eman, for your constant love, encouragement, and belief in me, even in the most challenging times.

Abstract

Global improvements in network infrastructure have enabled the development of exciting applications across a broad spectrum. These developments range from research powering lightweight educational, informative, and community-building services in rural and developing regions with poor internet accessibility and hardware to Collaborative Extended Reality applications that push the limits of the state-of-the-art network infrastructure available today with the aim of realizing ideas that, prior to recent advancements, were only available in fiction. Across the spectrum, significant challenges restrict the development and deployment of exciting new applications due to poor connectivity, limited access to high-performance devices, and unaffordable service costs in emerging regions and next-generation networked applications such as immersive reality and large-scale AI systems introducing unprecedented demands on bandwidth, latency, and sustainability in regions with state-of-the-art network infrastructure. This thesis addresses these twin challenges of digital inequality and network inefficiency by developing new systems and methodologies that operate across both the application and transport layers of the Internet stack.

In the first part of this dissertation, we present a series of lightweight web access systems designed for low-end phones, offline environments, and bandwidth-constrained regions. Through a global measurement study of 56 cities, we quantify disparities in page load times, web complexity, and mobile affordability. We then introduce Lite-Web, a browser-level rewriting system that accelerates existing websites on low-end devices. These web simplification efforts enabled internet accessibility in regions with poor internet accessibility and hardware constraints. However, many emerging regions suffer due to a lack of network infrastructure that creates a barrier between lightweight simplified webpages and people living in these regions. To address this, we also design Sonic, a novel hybrid system that leverages radio infrastructure to broadcast pre-rendered web content over FM radio and enable interaction through SMS, enabling access in disconnected regions such as rural Cameroon.

In the second part of the dissertation, we turn our focus to the transport layer, where emerging applications face severe limitations from current congestion control protocols. Using a new benchmarking framework, we evaluate the performance of state-of-the-art CCAs across synthetic and real 5G networks. Our analysis reveals significant mismatches between protocol behavior and the requirements of next-generation collaborative and immersive applications. To address this, we design Hera, a QoE-aware modular framework for next-generation immersive applications. By bridging the gap between application-level responsiveness and network-level adaptability, Hera lays the foundation for more scalable, robust, and high-fidelity multi-user immersive experiences.

Together, these contributions demonstrate how cross-layer design, from simplified content to smarter transport, can dramatically improve web accessibility, application quality of experience (QoE), and sustainability in both high-demand and underserved settings. The work advances a broader vision for an inclusive and efficient Internet: one that adapts to user constraints, application demands, and the infrastructural realities of the global majority.

Contents

List of Figures

List of Tables

1 Introduction

In this chapter, we present an overview of the dissertation, starting with the main motivations behind this work and the challenges in enabling efficient and equitable networked systems. Next, we outline the structure of the dissertation and highlight the main research contributions. Finally, we discuss the economic challenges that hinder the improvement of web access and connectivity in emerging regions.

1.1 Motivation

The Internet is one of the most transformative technologies of our time, enabling communication, education, commerce, and social participation on a global scale. Yet its benefits remain unevenly distributed [263]. While users in developed urban centers often enjoy fast, reliable, and ubiquitous connectivity, vast populations in emerging regions face persistent barriers that prevent them from fully participating in the digital world. At the same time, the Internet itself is evolving with bandwidth-intensive applications such as immersive media, large-scale AI services, and real-time collaboration tools that are pushing the limits of existing infrastructure [117, 238]. The central motivation behind this dissertation is to ensure that mobile users in emerging regions, from isolated rural areas to crowded urban hubs, can participate fully in the growing ecosystem of digital applications.

The first part focuses on rural and remote areas, especially across the Global South, where connectivity infrastructure is minimal or nonexistent. Geographic isolation, high deployment costs, and limited short-term economic incentives prevent network operators from extending service to these regions. Even basic services such as email, online search, or educational resources are inaccessible, creating an information gap that perpetuates social and economic disadvantage [197]. For these communities, the foremost challenge is establishing Internet connectivity in the face of significant economic and infrastructural barriers.

The second part focuses on areas where some form of connectivity exists, but true accessibility remains elusive. Networks are often unreliable, mobile data costs are prohibitive relative to income, and users frequently rely on low-end devices with limited processing power and storage. Meanwhile, much of the modern web is built for high-end smartphones and broadband networks, burdened by heavy JavaScript, high-resolution media, and resource-intensive scripts. This mismatch between content design and user capabilities results in long load times, degraded user experience, and, in many cases, complete inaccessibility to essential online resources [101].

Finally, the focus shifts to urban centers within emerging economies, where infrastructure is more advanced, but digital equity challenges persist. High population density, limited spectrum, and constrained backhaul capacity lead to severe contention for bandwidth, especially during peak usage hours. The growing adoption of real-time and data-intensive applications such as multi-party video conferencing, cloud gaming, and mixed reality exacerbates congestion [160]. These environments demand sophisticated transport-layer solutions that can manage shared resources efficiently without compromising the quality of experience for any one application or user.

This dissertation addresses these intertwined challenges by rethinking system design across both the application and transport layers. We develop new tools, protocols, and platforms that enable equitable web access and robust transport performance, particularly for users and applications operating in constrained environments.

1.2 Economics of Connectivity in Emerging Markets

Emerging regions present a complex set of economic and infrastructural challenges that significantly hinder the deployment and viability of compute-intensive AI services [120]. While these markets represent a large untapped user base, the cost structures and operational constraints differ fundamentally from those in developed economies, creating a persistent imbalance between service costs and user purchasing power. A critical challenge in emerging regions is that many areas lack the basic infrastructure to provide network connectivity.

In many of these regions, the absence of robust infrastructure is not merely a technical hurdle but a systemic limitation. Rural and peri-urban areas frequently suffer from unreliable or nonexistent electricity grids, which in turn undermines the viability of consistent Internet and compute service delivery. Telecommunications providers and cloud service operators face substantial capital expenditures to extend coverage to remote zones often through difficult terrain and with little expectation of short-term returns. As a result, these locations are deprioritized in favor of more densely populated or profitable urban centers. Moreover, factors such as political instability, logistical constraints, and limited public investment further exacerbate the slow pace of infrastructure development. From a commercial standpoint, the high cost of last-mile delivery, combined with the low average revenue per user (ARPU), makes infrastructure build-out economically unattractive to private sector players. This creates a structural disincentive to bridge the digital divide purely through market mechanisms.

In the second chapter of this thesis, we present a novel solution that leverages FM radio infrastructure to amplify web connectivity in emerging regions. We further discuss the performance of this system in real world settings.

1.3 Economics of Accessibility in Emerging Markets

One level above these regions are areas where basic infrastructure has been developed and made available to the public where we transition from the realm of Kilobits per second to Megabits per second. Unfortunately, the availability of basic infrastructure, in isolation, has proven to be insufficient to resolve the problem of web accessibility in emerging regions.

Bloated Web: Modern webpages are optimized for high-bandwidth environments and do not perform well in bandwidth-constrained networks. More accurately, modern webpages are unoptimized and bloated with resource-intensive Javascript [158], operating under the assumption that end users have access to the high-end hardware and network bandwidth necessary to access the content; a supposition that does not hold in emerging regions. This unnecessary webpage complexity can often be removed or simplified while maintaining extremely similar webpage quality with drastically improved performance [58, 106, 141].

Low-end hardware: Modern webpage complexity demands processing power that is often not supported by low-end mobile devices commonly used by people in emerging regions.

Low affordability: Recent research has emphasized that mere connectivity is not synonymous with accessibility. Qazi et al.[101] introduced the PAW (Price Adjusted Web access) metric, a framework that evaluates web affordability by accounting for disparities in income and broadband pricing. Their research reveals how modern webpages, optimized for high-bandwidth environments, disproportionately penalize users in bandwidth-constrained settings. In response, their Affordable Web For All (AW4A) initiative advocates for content simplification strategies, including image optimization and JavaScript minimization. More recently, semantic caching techniques leveraging Large Language Models (LLMs) have been proposed to enhance web affordability by reusing semantically similar content across pages, significantly reducing data transmission costs [14].

Despite the clear need for accessible web experiences in emerging markets, businesses are often not incentivized to optimize their web content for these regions. The dominant market logic prioritizes high-income users in developed economies, where faster networks and newer devices are the norm and generate the majority of advertising revenue and e-commerce transactions. Optimizing for low-end devices and constrained networks entails additional development costs with limited immediate financial return, making it an unattractive proposition for most companies. Moreover, without enforceable regulatory standards or widespread public pressure, accessibility for under-resourced regions is typically deprioritized in favor of features that serve affluent markets.

In the third chapter of this thesis, we present Liteweb, a solution to improving Web Accessibility in emerging regions by converting modern webpages into more accessible formats. We test this solution in emerging regions including rural Pakistan and evaluate the performance improvement achieved in these restrictive conditions.

1.4 Equitable Web Access in Emerging Markets

Finally, we explore emerging regions with access to modern network infrastructure. The goal of this research is to achieve *Equitable Web Access*; as exciting new applications in the worlds of AI and Extended Reality (XR) are being developed [117, 238], it is important to ensure that these applications are supported in emerging regions to prevent the digital divide from widening further. Equitable access to the internet and web in emerging markets is a multifaceted challenge that extends beyond infrastructure availability. True equity involves providing all individuals the capability to participate fully in the digital society. This requires not only physical access but also affordability, reliable and sufficient network speeds, relevant local content, digital literacy to be able to effectively use these services, appropriate hardware with sufficient compute power, meaningful use, safety, and empowerment.

In chapters 4 and 5 of this thesis, we discuss solutions to evaluate the performance of next generation applications in bandwidth-constrained environments. We study the gaps identified in our evaluation and design a modular framework to optimize the performance of next generation mobile applications in environments with high user numbers connected to the same mobile network base station competing for network resources.

1.5 The Next Billion Users Fallacy

Compared to traditional cloud services, AI applications such as large language models [44], real-time vision systems [200], and recommendation engines [70] demand significantly higher compute power, memory, and energy. As a result, the cost of delivering AI-based services is substantially higher and often prohibitive in regions where end-user pricing must be kept low for accessibility. In order to ensure equitable accessibility of new AI applications, economic viability is a core requirement for compute-intensive AI services in emerging regions. Unfortunately, in the current ecosystem, it is challenging for cloud compute services in these markets to be profitable due to a combination of high infrastructure costs, and limited consumer purchasing power [120].

This tension between high operational expenditure (OpEx) and constrained pricing leads to a structural imbalance. Emerging markets, which are often viewed as high-potential regions for future AI adoption, present businesses with both opportunity and risk. Although they offer access to vast new user bases, they require service providers to endure high customer acquisition costs and limited short-term revenue per user. Without a clear understanding of how infrastructure scale, utilization, and deployment strategies impact cost recovery, it is difficult to ensure viable AI expansion.

This thesis aims to address the key questions:

- How do high infrastructure and operational costs impact the ability of AI services to achieve economic viability in emerging regions?

- What pricing strategies are necessary to balance user adoption with cost recovery in these markets?

- Can AI services achieve profitability given the unique economic and infrastructural challenges of emerging regions?

To explore these questions, we develop the *Viability Calculator*, a modular framework that models the economics of AI service deployment in region-specific contexts. Our framework captures factors such as energy costs, PPP, infrastructure utilization, model efficiency, and demand growth. The framework estimates cost-per-query and breakeven conditions under realistic deployment scenarios. Using this system, we analyze a variety of AI service configurations across emerging regions. We evaluate the impact of different deployment strategies, and provide guidance for system designers and policymakers seeking to enable equitable, scalable, and viable AI infrastructure. Ultimately, our findings point to the need for tightly integrated solutions spanning hardware, software, and economic modeling to ensure the next generation of AI technology can reach underserved populations without sacrificing viability.

1.5.1 Economic Challenges

Emerging regions present a complex set of economic and infrastructural challenges that significantly hinder the deployment and viability of compute-intensive AI services [120]. While these markets represent a large untapped user base, the cost structures and operational constraints differ fundamentally from those in developed economies, creating a persistent imbalance between service costs and user purchasing power.

High Infrastructure Costs: The cost of building and maintaining AI infrastructure in emerging regions is disproportionately high. Import tariffs, limited local manufacturing capacity, and logistical expenses inflate the capital expenditure (CapEx) required to procure servers, GPUs, and cooling systems. Operational expenditure (OpEx) is further exacerbated by unreliable energy grids, which force operators to rely on expensive backup power solutions such as diesel generators or battery storage [12]. Additionally, power tariffs in many regions can be 2 to 3 times higher than global averages, making electricity the single largest component of recurring costs. The high power usage effectiveness (PUE) of older or climate-stressed facilities compounds these expenses.

Mismatch with Purchasing Power: Local purchasing power parity (PPP) in emerging regions is often too low to support the pricing models that AI services typically rely on in high-income markets [33]. While users in developed economies may tolerate subscription rates or per-query fees sufficient to cover infrastructure costs, similar price points would exclude most users in low-income communities. As a result, service providers are forced to operate at lower margins or even losses, betting on future scale and monetization.

Limited Access to Financing and Scale: Economies of scale are essential for amortizing the high upfront costs of AI deployments. However, emerging regions often lack both the market size and the financing mechanisms to support large-scale infrastructure rollouts. The absence of low-interest credit or public subsidies for AI infrastructure forces smaller or local players to depend on global cloud providers, which in turn impose higher per-query or per-hour fees.

Connectivity and Latency Constraints: Beyond hardware and energy costs, network limitations impose additional economic pressure. High-latency or unreliable internet connectivity can reduce server utilization and increase the overhead of distributed inference. Data egress fees from global cloud providers can further inflate costs, especially when services rely on frequent model updates or remote storage. Without localized caching or edge deployments, these overheads add a hidden but significant expense layer.

Regulatory and Policy Barriers: Regulatory environments in emerging regions are often underdeveloped or fragmented when it comes to cloud and AI services [241]. Import regulations, taxation, and inconsistent data localization policies introduce further complexity and cost. For example, restrictive data sovereignty requirements can force providers to build in-region infrastructure even when it is not economically optimal, leading to sub-scale deployments with poor cost efficiency.

1.5.2 Viability Calculator

To evaluate the economic viability of compute-intensive AI services in emerging markets, we design a modular simulation framework that models the financial and infrastructural footprint of AI deployment under region-specific constraints. The system takes as input a structured set of parameters that reflect both technical workload characteristics and economic conditions of the deployment environment. Its primary output consists of essential metrics including cost breakdowns, profitability estimates, and breakeven thresholds.

1.5.2.1 Input Parameters

The framework is designed to support flexible and realistic input modeling. Inputs are grouped into these categories:

Technical Configuration: Users specify the compute intensity of the AI model (e.g., FLOPs per inference), the number of inferences per month (e.g., based on daily query load), and hardware specifications such as GPU throughput (in TFLOPs), utilization rates, and power draw (in watts). These values allow the simulator to estimate raw infrastructure needs and power consumption.

Economic Context: Region-specific cost parameters are provided for electricity (USD/kWh), GPU rental or purchase cost (USD/hour), storage (USD/GB), bandwidth (USD/GB), and fixed operational overheads (USD/month). Additionally, users can specify revenue per user, either as a static value or estimated dynamically from purchasing power parity (PPP). These inputs enable localized cost modeling across multiple deployment environments.

Deployment Parameters: The framework accepts high-level deployment assumptions, such as the number of users, workload growth rate, redundancy requirements, and power usage effectiveness (PUE) of the infrastructure. These parameters influence both capital and operational scaling behavior over time.

1.5.2.2 Output Metrics

The system generates a set of interpretable metrics by combining compute workload characteristics, infrastructure parameters, and regional economic variables. We define these outputs formally below.

GPU Hours Given the total floating point operations F required monthly and GPU throughput T (in FLOPs/s) with utilization factor u, the required GPU hours H_{GPU} is:

$$H_{\text{GPU}} = \frac{F}{T \cdot u \cdot 3600} \tag{1.1}$$

Power consumption E (in kWh) accounts for power draw P (in watts) and power usage effectiveness (PUE) factor ϕ:

$$E = \left(\frac{P \cdot H_{\text{GPU}} \cdot 3600}{3.6 \times 10^6} \right) \cdot \phi \tag{1.2}$$

Monthly Service Cost. Let c_{gpu} be the hourly GPU rental cost, c_{elec} the cost per kWh, c_{ops} the base operational overhead, c_{maint} the fractional maintenance overhead, c_{store} the cost per GB of storage, and c_{net} the cost per GB of data transfer. The monthly total cost C_{total} is:

$$C_{\text{total}} = H_{\text{GPU}} \cdot c_{\text{gpu}} + E \cdot c_{\text{elec}} + c_{\text{maint}} \cdot (H_{\text{GPU}} \cdot c_{\text{gpu}}) + c_{\text{ops}} + D_{\text{store}} \cdot c_{\text{store}} + D_{\text{egress}} \cdot c_{\text{net}} \tag{1.3}$$

where D_{store} is the total storage required (GB), and D_{egress} is the total bandwidth (GB) transferred in and out.

Cost per Query. If the monthly number of queries is Q, the cost per query is:

$$C_{\text{query}} = \frac{C_{\text{total}}}{Q} \tag{1.4}$$

Revenue and Profit. Revenue is calculated based on the number of users U and estimated revenue per user r_u (often derived from PPP weighting). Total monthly revenue R is:

$$R = U \cdot r_u \tag{1.5}$$

Profit or loss is the difference between revenue and cost:

$$\Pi = R - C_{\text{total}} \tag{1.6}$$

Breakeven Threshold. Breakeven is achieved when profit $\Pi = 0$. Solving for the required user count $U_{\text{break-even}}$:

$$U_{\text{break-even}} = \frac{C_{\text{total}}}{r_u} \tag{1.7}$$

These equations form the core of the framework's cost modeling pipeline. They allow researchers to systematically explore the sensitivity of viability to energy pricing, model complexity, infrastructure scaling, and user monetization.

1.5.3 Analysis

Our framework is designed to help navigate a wide space of parameters that influence the economic viability of AI services. These parameters fall into two primary categories: (1) region-specific economic and infrastructure conditions, and (2) application-specific configuration and workload requirements. In this section, we focus on the former by fixing the model and hardware setup while varying regional economic inputs to analyze their effect on viability. The results show how regional disparities shape breakeven conditions and cost-efficiency for AI deployments. In future work, the same methodology can be applied to the second category to explore different model architectures, utilization levels, and deployment modes.

1.5.3.1 Application Profiles and Hardware Demands

Application	Est. Training FLOPs (TFLOPs)	Approx. Inference FLOPs (TFLOPs)	Typical Hardware	USA Cost ($)	Kenya Cost ($)	India Cost ($)
Image Captioning	50,000,000	0.8	T4 / L4 GPU	375.00	504.00	420.00
Real-time Translation	100,000,000	2.0	A10G / A100	937.50	1,260.00	1,050.00
Video Moderation	300,000,000	3.5	A100 + NVLink	1,640.63	2,205.00	1,837.50
Voice-to-Text Transcription	50,000,000	1.2	CPU + GPU hybrid	562.50	756.00	630.00
Recommendation Systems	10,000,000	0.5	CPU+Memory-intensive	234.38	315.00	262.50

Table 1.1: Example AI Applications, Estimated Training and Inference Complexity, and Monthly Service Costs by Region

Economic viability is largely shaped by the nature of the application itself. Different AI use cases vary widely in their inference complexity, throughput requirements, and infrastructure utilization profiles. Table 1.1 summarizes representative AI applications and the types of hardware and compute requirements they typically demand. This illustrates how different points in the workload parameter space can be simulated using our framework to assess feasibility under multiple configurations.

1.5.3.2 Regional Variation in Breakeven Volume

To evaluate when AI services become profitable in different regional markets, we simulate monthly profit across a range of user volumes while holding the AI model architecture and infrastructure setup fixed. The total cost of service includes cloud GPU usage, energy consumption, maintenance overhead, storage, bandwidth, and CloudSQL database access costs. each of which scales up with user count. As the number of users increases, compute demand, data transfer volume, and database interactions rise accordingly, placing upward pressure on infrastructure costs.

Figure 1.1 illustrates how profitability trajectories vary significantly across countries. India achieves profitability relatively early, benefiting from a combination of moderate electricity rates, affordable cloud compute, and higher PPP-adjusted revenue per user. In contrast, Pakistan,

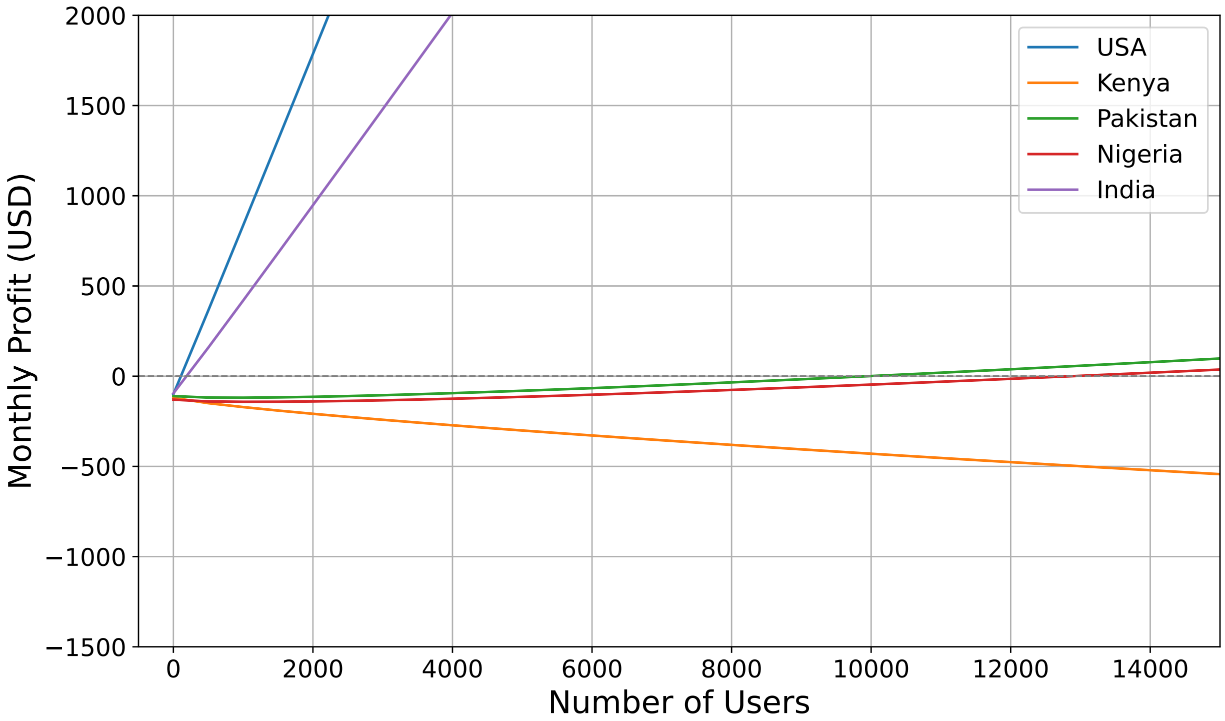


Figure 1.1: Monthly profit vs. user volume across regions

Kenya, and Nigeria struggle to cross the breakeven point even as user counts rise, largely due to lower per-user revenue and higher marginal costs associated with GPU rentals, networking, and backend services like CloudSQL. Notably, although Kenya and Nigeria are geographically close, differences in electricity pricing and cloud infrastructure access lead to divergent cost curves, underscoring the importance of country-specific infrastructure and policy contexts.

1.5.3.3 Cost Analysis for AI Labeling at Scale

To analyze the cost of running an AI-powered labeling service, we simulate the monthly infrastructure expenditure for processing 8,000 high-resolution videos using inference on GPU-backed infrastructure. Inputs to this simulation include cloud GPU pricing from the Google Cloud Pricing Calculator [64], region-specific energy costs [203], and standard storage and bandwidth charges. As shown in Table 1.2, operational costs for AI-based labeling in emerging regions

Country	Total Monthly Service Cost	Labeler Wage
USA	509.92	5583.00
Kenya	607.32	300.00
Pakistan	553.81	211.41
Nigeria	615.94	121.00
India	538.10	280.00

Table 1.2: Monthly AI Labeling Cost vs. Human Labeler Wage

typically exceed $500 per month. In contrast, human labeler wages in these same regions range from $121 in Nigeria to around $300 in Kenya.

This stark disparity reveals a clear cost asymmetry. For many non-real-time applications, it is still cheaper to rely on human annotation. For example, if a labeler annotates 400 videos per month, just 20 labelers would be sufficient to handle the same workload, at a fraction of the infrastructure cost. Given the high infrastructure costs and relatively low wages in emerging regions, AI-only labeling pipelines are economically suboptimal unless amortized across significantly higher volumes or combined with human-in-the-loop verification. This finding supports the growing trend toward hybrid human-AI annotation workflows and points to the need for task-aware deployment strategies that adapt based on regional labor and compute economics.

1.5.3.4 Key Takeaways

Together, these experiments demonstrate the importance of navigating a complex and multidimensional parameter space when evaluating the viability of AI services in emerging regions. By separating the analysis into regional economic parameters and application-specific workload configurations we reveal how both structural costs and deployment scenarios influence economic feasibility. The user volume analysis highlights that countries geographically close to each other or within similar economic bands often face comparable cost pressures due to shared infrastructure limitations, similar energy pricing structures, and constrained purchasing power. In contrast, India, while still considered an emerging market, benefits from lower cloud pricing and improved infrastructure maturity, enabling lower breakeven thresholds. These regional similarities and differences are essential for guiding deployment strategies. A single model of cost or revenue cannot be applied uniformly across frontier markets.

Moreover, the geographic distribution of potential users must inform any global AI expansion strategy. Southeast and East Asia alone account for over 30% of the world's population, with countries like India, Indonesia, Bangladesh, and the Philippines representing billions of users with growing digital footprints. Designing systems that are economically viable in this region could unlock immense scale and impact. Conversely, low-population or high-cost regions may require smaller-scale, subsidized, or hybrid deployments to be viable. Similarly, our task-based cost simulations show that compute and storage requirements vary significantly across applications further motivating custom infrastructure provisioning strategies. Ultimately, the modular framework we present enables detailed exploration of this full design space. It empowers researchers, practitioners, and policymakers to identify the configurations where AI services can be viably deployed and scale in alignment with local economics and population dynamics. As AI systems expand globally, such localized planning will be key to both equitable access and long-term operational viability.

1.6 Conclusion

In this analysis, we discuss the inevitable deadlock that cloud compute services are approaching in emerging regions between the increasing demand for compute-intensive AI services and the fragile economics of deploying infrastructure in low-income markets. We show that several compounding factors make economic viability difficult including high infrastructure and energy costs, limited grid reliability, and lower PPP. To address this, we present a modular simulation framework that allows stakeholders to model the lifecycle costs of AI infrastructure under regional constraints and assess breakeven thresholds. Our framework integrates demand forecasting, infrastructure sizing, model efficiency, and operational factors to quantify the cost per query and explore the impact of different optimizations. Our findings emphasize the importance of holistic cost modeling and design-aware deployment strategies. The goal of this framework is to help enable future research, tools, and collaborative efforts that drive the development of equitable and viable AI.

1.7 Dissertation Overview

The thesis is organized into two major parts:

- **Part I: Addressing Digital Inequality through Web Access Innovation.** We begin by quantifying the global digital divide through a measurement study and then develop a suite of systems, including Lite-Web and SONIC, to make web content more accessible for low-end devices and offline users.

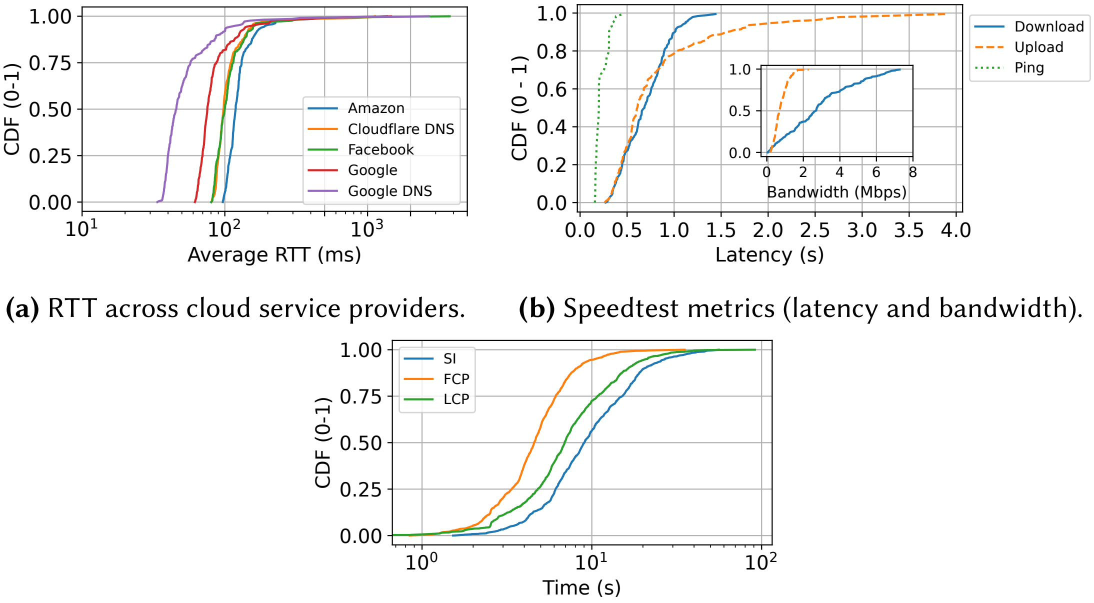
- **Part II: Optimizing Transport Protocols for Emerging Applications.** We analyze the performance of modern congestion control protocols under real and synthetic 5G conditions using the Zeus benchmarking framework, and propose Hera, a delay-aware, application-informed congestion control algorithm optimized for immersive applications.

2 Tuning Into The Web

This chapter is adapted from the preprint version of "SONIC: Cost-Effective Web Access for Developing Countries" and "SONIC: Connect the Unconnected via FM Radio & SMS", published in ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT) [187]. In this chapter, we introduce a novel solution that leverages radio signals to improve web accessibility in emerging regions.

ABSTRACT

Over 2.6 billion people remain without access to the Internet in 2025. This phenomenon is especially pronounced in developing regions, where cost and infrastructure limitations are major barriers to connectivity. In response, we design Sonic, a low-cost, scalable data delivery system that builds on existing infrastructures: FM radio for downlink broadcasting, and SMS for personalized uplink. Sonic is motivated by the widespread availability of FM radio and SMS infrastructure in developing regions, along with embedded FM radio tuners in affordable mobile phones. Sonic offers several innovations to effectively transmit Web content over sound over FM radio, in a reliable and compressed form. For example, we transmit pre-rendered webpages and leverage pixel interpolation to recover errors at the receiver. We further modify Android to offer a simpler deployment pipeline, supporting a wide range of devices. We deployed Sonic at an FM radio station in Cameroon for six weeks with 30 participants. Our results demonstrate a sustained



(a) RTT across cloud service providers.

(b) Speedtest metrics (latency and bandwidth).

(c) Web performance metrics (SI, FCP, LCP).

Figure 2.1: Network performance of MTN, the best available mobile network provider at our deployment site in Cameroon.

downlink throughput of 10 kbps, less than 20% loss for a majority of transmissions with signal strength above -90 dbM, and a strong user engagement across both Web browsing and ChatGPT interactions.

2.1 Introduction

The internet has fundamentally reshaped societies worldwide, driving economic growth, fostering new industries, and becoming indispensable in education, healthcare, and employment. Yet, despite its profound influence, 32.4%, over 2.6 billion people remain disconnected in 2025 [224]. This *digital divide* is not only due to a lack of infrastructure. According to GSMA Intelligence [96], 3.1 billion people live within the coverage area of a mobile broadband network but remain disconnected because they cannot afford the necessary devices and data plans to get online.

The offline population is especially concentrated in developing regions. For example, India has 651 million people (44.7%) offline, Pakistan 137 million (54.3%), and Nigeria 128 million (54.6%) [224]. In the Central African Republic and South Sudan, over 84% of the population remains without internet access [224]. The consequences of being "unconnected" go beyond missed economic opportunities; it severely limits access to essential services such as education, healthcare, and financial resources, further exacerbating existing inequalities.

Several initiatives have attempted to bridge this gap. Starlink [227] provides global coverage through low earth orbit (LEO) satellites. Google's Project Loon [84] sought to deliver internet via high-altitude balloons, while Facebook's Aquila [111] aimed to use solar-powered drones (both ultimately discontinued due to maintenance costs and scalability challenges). Project Taara [234] is a recent initiative that transmits data using laser beams over long distances (20 km) at high speeds (20 Gbps). Unfortunately, these efforts remain prohibitively expensive in most developing regions. For example, in Zambia, classified as one of the UN's least developed countries, Starlink's $40 monthly subscription (plus $180 hardware fee) is prohibitively expensive compared to the country's $108 monthly GNI per capita.

In this chapter, we build on the preliminary work by [188] which introduced an initial framework to transmit simplified webpages over FM radio. Webpages are pre-rendered as images which are then modulated into audio signals and transmitted via FM radio. We advance [188] to Sonic as follows:

System Architecture. We design the full software architecture of the Sonic server and Android client, which includes modules for content rendering, prefetching, encoding, transmission, decoding, and error correction.

FM Tuning on Android. We leverage the internal FM tuner found in many Android smartphones to receive and decode Sonic transmissions. We do this by modifying LineageOS, an open-source Android operating system, to enable programmatic access to the FM hardware, allowing other apps to control FM chip tuning and access the received audio stream directly, without needing to root the device.

LLM Support. We extend Sonic's functionality to support LLM interactions over FM radio. We use the same audio modulation pipeline to allow users to query models like ChatGPT via SMS and receive coherent responses without internet access. We show that LLM responses, due to their smaller size, can be transmitted significantly faster than full webpages.

Real-world Deployment. We report on a six-week deployment at a live FM radio station in Cameroon. During this period, 30 participants used the Sonic app to request webpages and interact with ChatGPT via SMS. Our evaluation shows that Sonic can sustain a transmission rate of 10 kbps, with stable reception achievable at a Received Signal Strength Indicator (RSSI) value up to -90 dBM. Mean decoding accuracy remained at 71% under real-world conditions.

2.2 Motivation

Affordability and QoE in Rural Areas. Mobile internet adoption is rapidly increasing in low- and middle-income countries [96], yet ensuring affordability and a good quality of experience (QoE) remains a challenge. In many remote areas, even when mobile broadband is available, data costs are prohibitively high relative to average income levels. To assess affordability and QoE in a rural setting, we conduct network performance measurements and a survey at the Sonic deployment site in Cameroon.

We select MTN [171], the best available mobile network provider in the area, and purchase its monthly data plan, which offers 9.2 GB for $14. Following the methodology in [244], we use `mtr` [137], a tool for measuring latency and tracing network paths to examine routing changes and packet losses. Additionally, we employ `Speedtest CLI` [180], a command-line tool for measuring latency, download, and upload speeds. We also evaluate web performance using `Google Lighthouse` [150], an automated tool that provides key website performance metrics, including Speed Index (SI), First Contentful Paint (FCP), and Largest Contentful Paint (LCP). Given the data limits of our mobile internet plan, we conduct `mtr` and `Speedtest` measurements once every

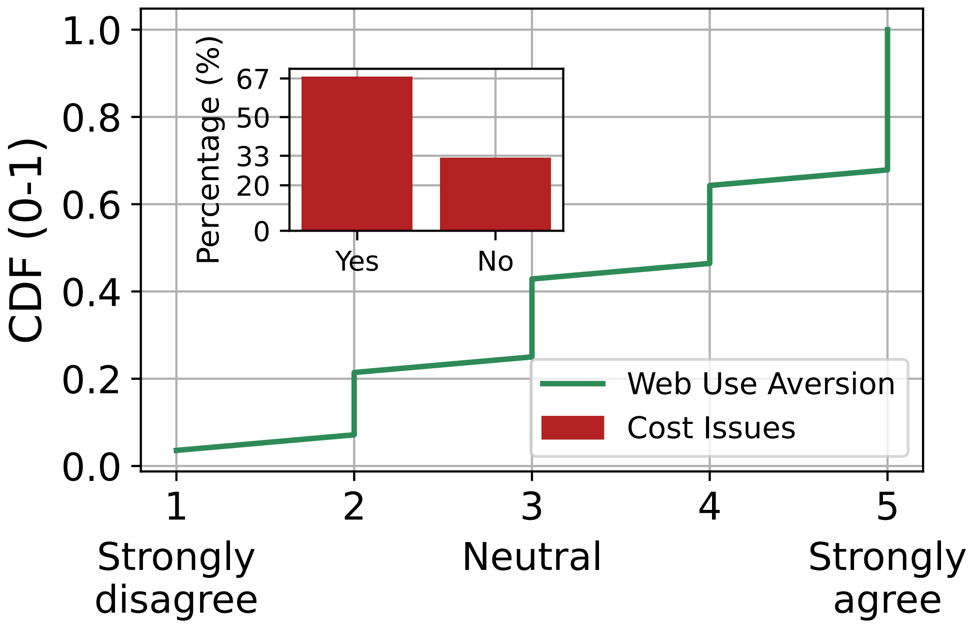


Figure 2.2: Barriers to Web access from speed and cost. The main plot shows CDF of how Sonic users agree with avoiding websites due to slow internet. The inset shows the % of users indicating that their browsing is restricted by data costs.

three hours throughout our deployment. For Lighthouse evaluations, we rely on data collected in the wild as a response to real Sonic users (see Table 2.1).

Figure 2.1(a) shows the cumulative distribution function (CDF) of average round-trip time (RTT) towards popular content providers (Amazon, Facebook, and Google), and DNS operators (Cloudflare and Google). The figure shows RTTs higher than 100ms for most measurements and providers, with the exception of Google. As observed in [243], this is due to Google footprint in Africa which was also confirmed by our path analysis.

Next, Figure 2.1(b) summarizes the speedtest analysis. With respect to the RTT to OOKLA servers (ping), the figure shows a similar trend as Figure 2.1(a). The figure further shows RTT under load, i.e. while measuring both download and upload speeds, showing a 3.4x growth (from 200 to 680 ms, at the median) thus suggesting large buffer in use (a phenomenon typically called bufferbloat [88]). Despite these large buffers, users experience a median download speed of 2.6 Mbps and an upload speed of 0.69 Mbps—both drastically below the global average—at a mobile internet price comparable to developed nations ($1.5 per GB) [244]. At these speeds, web performance is extremely affected, as visualized in Figure 2.1(c). The median SI is at 9 seconds, FCP at 4.5 seconds and LCP at 6.9 seconds, significantly slower than the web vitals threshold (LCP < 2.5 seconds) recommended by Google [163].

The combination of high latencies and low download speeds can frustrate users and discourage regular internet use. Figure 2.2 presents responses to two survey questions assessing barriers to web access at our deployment site in Cameroon. Survey participants are 30 Cameroonians who participated in Sonic deployment. The main plot shows the CDF of web use aversion, based on responses to the question: *"To what extent do you agree with the following statement: 'I occasionally avoid visiting certain websites because my Internet is too slow to load them.'"* (1 = Strongly disagree, 5 = Strongly agree). Over 75% of the participants indicate either neutrality or agreement with avoiding websites due to slow internet speeds. The inset plot shows responses to the binary question: *"Is your web browsing experience often restricted by data costs?"* where 67% of users answered "Yes."

These results sheds light on the affordability of web access in low-income regions like Cameroon. Beyond slow speeds, high data costs severely limit both how often and how effectively users can engage with the internet, restricting access to information that many take for granted. Indeed, we also asked participants if they had heard of ChatGPT prior to participating in the experiment. 80% reported "No", revealing a broader lack of exposure to transformative (and popular) technologies simply because the current infrastructure never allowed these tools to reach them.

FM Radio Availability. FM radio still remains widely used in developing regions. In a study covering 39 countries in Africa, about 65% of adults reported listening to radio at least a few times per week, with no major difference in the rural-urban gap in radio access [11]. A recent case study in northern Ghana found FM radio to be the most reliable and trusted source of developmental information in rural communities, providing vital content on agriculture, education, and health in local languages [20].

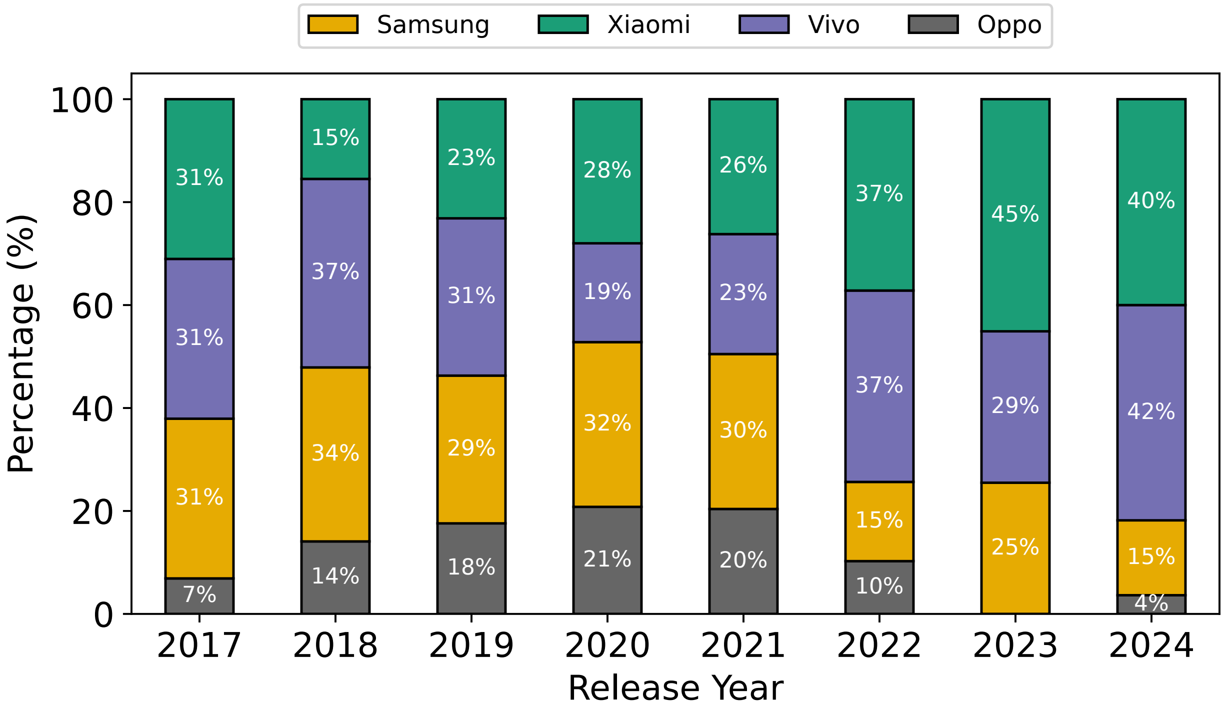


Figure 2.3: Mobile phones supporting FM receivers for the top four Android brands grouped by release year (2017-2024).

Technical studies confirm that FM signals reliably cover large areas, though terrain can influence signal quality. In Nigeria, measurements around a 20 kW FM station showed stable reception up to 50 km, beyond which quality declined due to sandy or silty soil and obstructed line-of-sight [25]. A study from Nepal emphasized the importance of antenna height and placement in extending coverage, even in hilly regions with shadow zones between elevations [32]. Despite geographic challenges, FM radio continues to provide consistent, low-cost coverage in most rural areas, with signal strength sufficient for everyday use.

Building on the widespread availability of FM radio connectivity, we next investigate the landscape of FM radio support in mobile phones. Specifically, we analyzed the prevalence of FM receivers in Android smartphones currently available on the market. To do this, we scraped the database of mobile phone specifications from GSMArena [98], identifying which models support FM radio and recording their release years and market prices. We then cross-referenced this

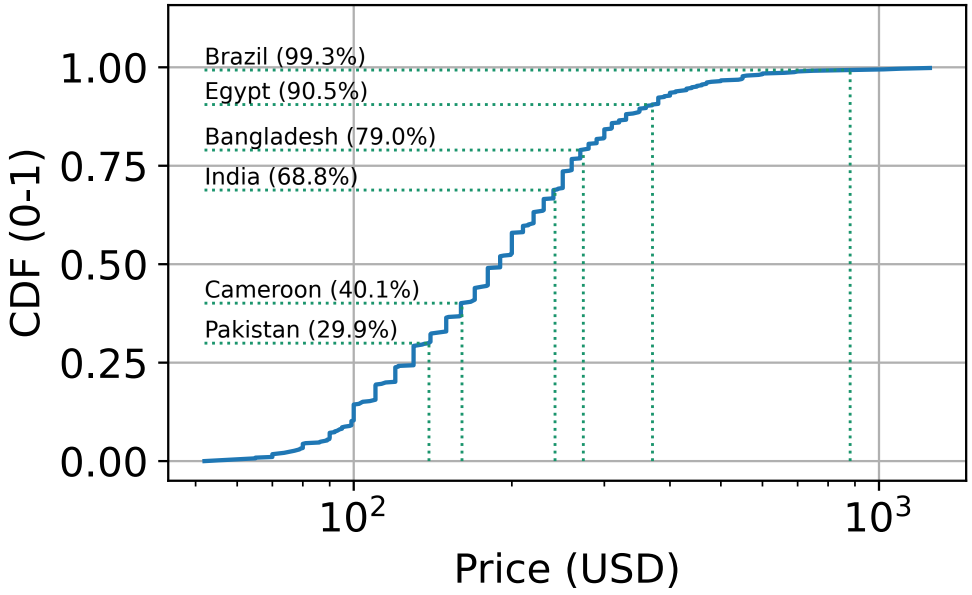


Figure 2.4: CDF of prices for android phones equipped with an FM radio receiver. Green dotted lines indicate the monthly Gross National Income (GNI) per capita for selected low- and middle-income countries.

information with global economic statistics [259] to assess the practical reach of FM-based data reception.

Figure 2.3 shows the distribution of FM-capable phones released by the top four Android smartphone brands [69] over the past eight years. Despite a gradual decline in newer models, FM radio functionality remains prevalent—especially among Xiaomi and Vivo devices, where approximately 40% of current models still include FM support. In total, these top four brands alone account for 571 FM-capable models, highlighting a significant and readily accessible user base for FM-based data delivery.

We next evaluate the affordability of these 571 FM-capable phone models in low-income countries, using monthly gross national income (GNI) per capita as a benchmark [259]. Figure 2.4 shows the price distribution of these phones (blue line) alongside GNI thresholds (green dotted lines) for six developing countries: Pakistan, Cameroon, India, Bangladesh, Egypt, and Brazil. The figure shows that a substantial share of devices are within reach of average consumers, from

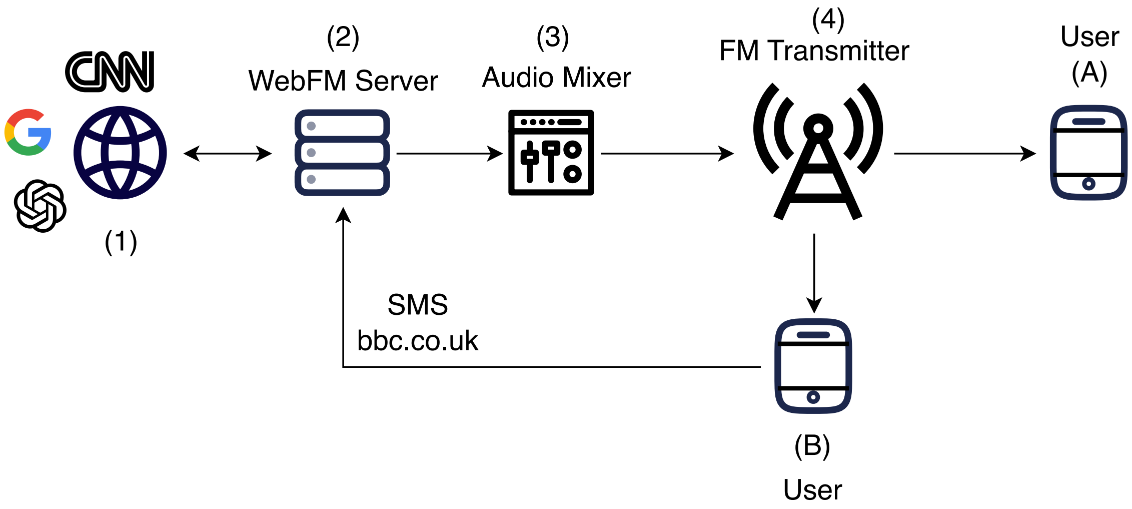


Figure 2.5: Sonic workflow.

a minimum of 30% in Pakistan up to 99.3% in Brazil. These results highlight strong market availability of FM-enabled phones at accessible price points in low and middle-income regions.

2.3 Sonic

Figure 2.5 shows how Sonic operates. Users A and B both have the Sonic app installed on their smartphones, equipped with chipsets featuring FM receivers. However, only User B's device supports SMS. The Sonic app runs on both smartphones with FM receivers tuned to a specific frequency to decode incoming data-over-audio streams.

User B requests a webpage using the Sonic app, which sends an SMS containing the desired URL (e.g. `bbc.co.uk`) to the Sonic server. The server listens for incoming SMS messages, retrieves the webpage, captures a screenshot of the rendered page, and compresses it into a WebP image. The image is then encoded into sound and broadcasted via an FM transmitter. Both User A and User B receive the transmitted webpage on their smartphones. In the following, we describe each Sonic component which enables the above workflow.

2.3.1 Sonic Server

Figure 2.6 shows the architecture of the Sonic server. The server runs as a Docker [78] container on a computer located at a radio station. It consists of several key components that work together to process user requests. The SMS Manager handles incoming messages, while the Screenshot Queue processes webpage URLs in a First Come, First Serve (FCFS) manner. A Cache stores recently requested URLs to avoid redundant processing. The Encoder converts responses into Sonic *file format* (see Section 2.3.1.1) and encodes them to audio. Finally, the Player Queue manages the order in which the encoded audio files are played, also following the FCFS policy.

The SMS Manager continuously listens for SMS messages using a USB mobile dongle. These messages are received on a phone number assigned to the SIM card inserted into the dongle. Messages sent by the Sonic app contain the sender's information in the headers and a payload formatted as: `<type> <body>` (e.g. `url https://nytimes.com`). When a new message is received, the server classifies it as either an LLM prompt or a webpage URL depending on the `<type>` identifier.

If the message contains a URL, it is added to the Screenshot Queue. The system checks whether this URL has been retrieved within the current transmission window using the Cache. If not, the Screenshot Queue utilizes Selenium [220] to load the page in Google Chrome with a mobile resolution of an iPhone SE device (375 × 667 pixels). Once the webpage is fully loaded, a full-page screenshot is taken and resized to a width of 320 pixels. We empirically selected 320 pixels as a sweet spot where both the content layout and text remain comfortably readable to the human eye. The screenshot is then encoded into a Sonic file format and subsequently converted into audio using the `Quiet` [194] library at the Encoder. The resulting audio is then added to the Player Queue. In case of cached requests, the cached response directly moves to the Player Queue.

If the message contains an LLM prompt, the server makes an LLM inference API call to either

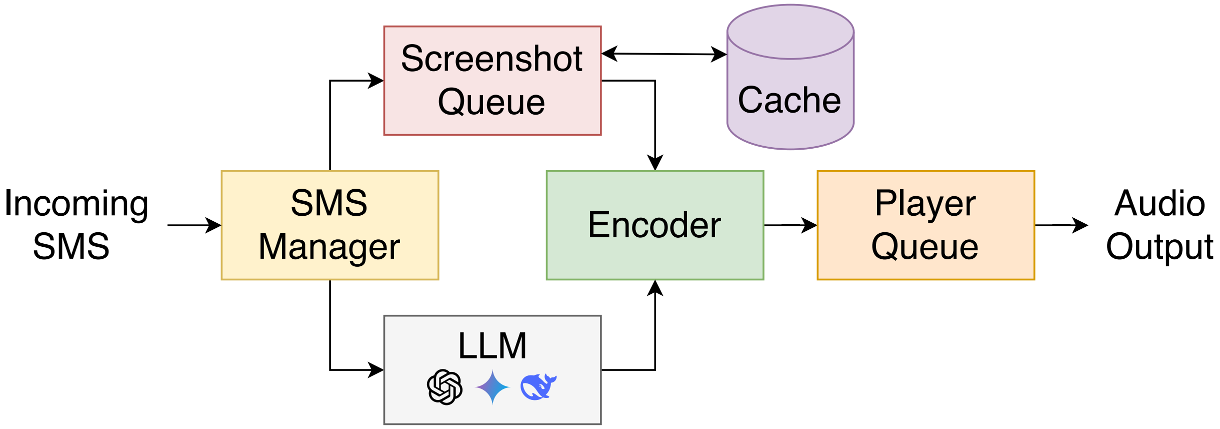


Figure 2.6: Sonic server architecture.

a locally running LLM, or a cloud API (e.g. `OpenAI Chat Completions API` [181]). The response from the LLM is encoded into the Sonic file format containing structured metadata and payload. Finally, the server generates a corresponding audio using the `Quiet` library, which is then added to the Player Queue.

2.3.1.1 Encoding

Encoding takes place in the Encoder, which consists of two steps: 1) encoding responses to a new file format, referred to as a *webfm file*; and 2) converting this webfm file into a waveform audio file (WAV). The resulting audio file is then added to the Player Queue.

Sonic File Format. Sonic uses a new file format that allows the decoder to easily distinguish specific parts of the transmission in absence of a continuous uplink. As illustrated in Figure 2.7, this format includes intermediate headers (such as "MDTA," "LNKS," "SDTA," etc.) that separate internal sections within the metadata and payload. This structure helps reconstruct content in cases when metadata is fully received but only parts of the payload are received properly. Additionally, each payload frame is prefixed with "C137," inspired by *Rick and Morty*'s C-137 dimension [204], which helps distinguish the start point of each frame. Furthermore, "C137" serves as a keepalive message, allowing the app to notify users that the server is online when transmissions are in progress.

Webpage Compression. Unlike encoding LLM interactions, where the payload is simply appended to metadata, encoding webpages presents a significant challenge due to the limited data rates achievable over audio transmission—typically only tens of kbps (see Section 5.2). According to [252], the average mobile webpage size is approximately 2 MB. Broadcasting such a page via Sonic could take tens of minutes. Moreover, devices experiencing poor RSSI may struggle to reconstruct the page, as critical web components, such as JavaScript, may fail in the presence of unrecoverable errors.

To address these challenges, we must: 1) significantly compress webpages, and 2) ensure resilience against noise. Various methods exist for reducing webpage sizes, such as compression proxies [93, 153], reader modes [91, 213], JavaScript cleaners [54, 55, 58], and redundant code removal [141, 158]. These approaches remain vulnerable to noise and require extensive forward error correction (FEC), necessitating a system design that accounts for the worst-case receiver conditions.

Instead, we utilize a solution where performance degrades gracefully as a function of the receiver's RSSI, analogous to how audio quality deteriorates under poor reception. Inspired by [38, 39], which demonstrate how image quality over RDS degrades with RSSI, we opt to transmit images of rendered webpages rather than raw web files (HTML/JavaScript/CSS). This approach provides both compression and resilience: a 2 MB webpage can be compressed into a few hundred KB, and images remain interpretable even if some pixels are lost.

Interactivity. Modern webpages enable user interaction via hyperlinks, menus, and search boxes, whereas images are inherently static. To introduce interactivity, [36] proposes *click maps*, which store <x, y> coordinates of interactive elements. We adopt this approach, allowing Sonic to notify the server (via SMS, if available) when a user clicks on a coordinate, retrieving the corresponding page if it is not already cached. Given Sonic's potentially slow network conditions–seconds for uplink and minutes for downlink–we limit interactivity to hyperlinks.

Image Format. Unlike [36], which requires lossless PNG for crowdsourced screenshot merging, Sonic utilizes *WebP* [17], a modern format offering superior compression. Webpages are captured as WebP images at 10% quality, significantly reducing file size while maintaining readability. Images are 320 pixels wide and up to 10,000 pixels tall, enabling users to *scroll* with minimal data overhead. To accommodate different screen sizes, images are resized using a scaling factor (`screen width / 320`), ensuring accurate click map coordinates.

Modulation. We use the `Quiet` library to modulate Sonic files as waveform audio files (WAV). Inspired by "audible-7k-channel", we created a new modulation profile that uses Orthogonal Frequency-Division Multiplexing (OFDM)—a multi-carrier modulation technique that divides the available spectrum into multiple orthogonal narrow-band signals called sub-carriers. Our profile uses 92 sub-carriers, with a center frequency of 9.2 KHz, achieving a rate of 10 kbps.

2.3.1.2 Pushing

During low-usage periods, typically at night, Sonic *pushes* [232] popular webpages to its clients to ensure faster response times during peak hours. For each webpage, Sonic also pushes some internal links allowing users to seamlessly interact with a webpage without delay. However, pushing every hyperlink on a webpage would overwhelm the system's limited transmission bandwidth. To address this, we use a prioritization metric to rank internal links based on their importance:

$$score = 0.68 \cdot w \cdot h - 0.32 \cdot y$$

where `w`, `h`, and `y` denote the width, height, and vertical distance from the top of the page, respectively. We derive this metric from a Prolific user study detailed in Section 2.4.

When a webpage enters the Screenshot Queue, Sonic computes this score for each hyperlink and selects the top three ranked links. These are added to a separate idle queue that is only activated when the server has no active transmissions. As Sonic primarily targets informative sites like news and blogs, this metric prioritizes pages with top headlines, larger images, and prominent font sizes, ensuring that key content is already available when users attempt to follow links. Additionally, it penalizes links that are vertically farther from the top of the page.

2.3.2 Sonic Client

At a minimum, Sonic users require a smartphone with a built-in FM radio receiver, serving as the *downlink*, along with a wired earphone to act as an antenna. Additionally, users who wish to send requests, such as retrieving a webpage or interacting with the LLM, need access to an SMS service for the *uplink*. On the software side, Sonic operates as a user-space application on a modified version of Android (see Figure 2.8). We detail the Sonic client in the following.

OS Integration. The Sonic client relies on FM radio hardware to receive data transmissions, but modern Android devices do not expose FM chip access to third-party applications. Default FM radio apps are shipped as system apps, integrated into the ROM and signed with privileged keys that allow hardware-level access. Apps like Sonic are unable to access FM audio without rooting the device and allowing superuser access, which is unrealistic for adoption and raises significant security concerns. As such, enabling FM-based decoding requires changes at the operating system level.

We build a proof-of-concept implementation based on LineageOS [151], a widely supported open-source Android distribution. We modify its default FM radio app to: 1) allow tuning the FM chip to a specific frequency, and 2) forward decoded audio streams to other apps like Sonic without needing root or elevated permissions. This is achieved by implementing a `BroadcastReceiver` [19] in the Sonic app and a matching sender in the default FM radio app that transmits raw audio buffers in real time. This change allows Sonic to passively listen to FM broadcasts and decode data as it arrives. We verified this approach by flashing our customized LineageOS build onto devices that use Qualcomm Snapdragon chipsets, including Xiaomi Redmi Go. We observe that as long as FM radio drivers are available, this approach can be extended to any other chipset. Manufacturers would implement this change when building the stock operating system for devices they ship with FM radio support (see Section 2.3.3).

User Interface. Figure 2.9 shows the Sonic app's user interface (UI), which consists of three sections: Browser, ChatGPT and Knowledge Hub. The Browser section mimics modern web browsers. It features a search bar at the top, allowing users to request URLs. A list of requested URLs is also displayed, and once received, they appear as "ready to view." The ChatGPT section provides a chat-like interface where users can interact with ChatGPT by asking questions and receiving responses. Since FM radio operates as a broadcast system, all devices tuned to the same frequency receive the same content, even if it wasn't specifically requested by those devices. This broadcast nature of FM radio is leveraged by Sonic with its Knowledge Hub section. In this section, users can access a list of webpages and ChatGPT responses that are popular within their region, allowing them to discover trending content shared by others nearby.

Background Service. In addition to the UI, the Sonic app runs a background service that performs two key functions: 1) continuously listening to FM radio audio streams at a specific frequency, and 2) decoding transmissions when a Sonic-encoded signal is detected. The background service listens to bytes broadcasted by the default FM radio app using a `BroadcastReceiver`. To decode Sonic-encoded transmissions, it uses a modified version of `Quiet`'s Android library [18], which by default decodes audio from the device's microphone. We modified the library to accept bytes retrieved from the phone's FM radio app and decode them using the same modulation profile described in Section 2.3.1.1.

This background service operates independently of the user interface and automatically stores all incoming transmissions in an SQLite database. To optimize storage, any unaccessed content is automatically deleted after one day.

Error Correction. We utilize crc32 checksums per frame to detect errors. Furthermore, an inner FEC scheme (v29) and an outer FEC scheme (rs8) are used to correct transmission errors. For completely lost frames, the Sonic receiver applies nearest-neighbor pixel interpolation [211], replacing missing pixels with the value of their adjacent left pixel given that webpage consists mostly of text read from left to right.

In the literature, better-performing techniques exist to recover missing pixels in images [53, 193, 226], leveraging deep neural networks to learn patterns and structures of the image or utilizing sparsity and gradients in the data to fill in the missing regions. These techniques are both memory- and CPU-intensive, far beyond what a low-end mobile device can support today. Thus, we adopt a lightweight approach proposed and benchmarked by [188] that provides consistently high content readability scores even at a 20% pixel loss rate.

2.3.3 Discussion

Rollout. Governments, NGOs, and other organizations can roll out Sonic across a wide range of Android phones by pre-installing a modified version of the LineageOS ROM (detailed in Section 2.3.2). Moreover, smartphone manufacturers could adopt this approach natively when designing their operating systems. Manufacturers can enable support for data-over-FM use cases like Sonic without compromising system security or requiring root access by bundling a modified FM radio app that exposes decoded audio streams to other applications. This would allow future devices to support FM-based services out of the box with minimal engineering overhead.

Incentives and Monetization. Sonic users benefit by gaining access to a streamlined version of the Web in areas where such access is typically unavailable. For providers, one approach is to charge users directly. However, this can be difficult since users receiving content via downlink are passive, making it hard to know when or if content is being accessed. As an alternative, providers could link the service to SMS, allowing paying users to request content on demand, while keeping access free for others. Notably, FM broadcasting costs remain constant, regardless of the number of listeners.

A more promising revenue model mirrors how traditional radio stations function: expanding the audience to increase advertising revenue. Sonic adds a unique offering that could draw more users, potentially enhancing ad-based profits. Furthermore, ads are no longer limited to audio—they can now include visuals embedded in the web pages.

Privacy Concerns. At a high level, Sonic resembles acceleration platforms like Google AMP [93] and WebLight [153], which modify webpage content before sending it to users. These services typically rely on access to both the URLs users request and the content they consume, which raises potential privacy concerns. While a Sonic server could, in principle, gather enough information to build user profiles, it avoids this issue by using FM radio as a broadcast channel. This mode of delivery makes it impossible to identify who is receiving the content. As a result, users on the downlink side remain fully anonymous, passively receiving data initiated by others nearby, without any associated privacy risk.

Limitations. Sonic does not enable access to login-restricted content, such as online banking or social media accounts. This is unfeasible for downlink-only users (i.e. no SMS support); for uplink users, it would involve sharing login credentials with the Sonic server which is a significant privacy risk. Moreover, because content is broadcasted, any personalized information (like account details) would be exposed to anyone within range, further compromising privacy.

Next, Sonic lacks support for video which is a major part of modern web usage, e.g. streaming, news, and social media. Sonic's limited bandwidth makes video streaming infeasible. Instead, video content is replaced with static, non-interactive thumbnails. Likewise, advanced features driven by JavaScript or CSS are not supported, as Sonic only transmits simplified, pre-rendered versions of webpages.

2.4 Benchmarking

This section benchmarks Sonic under controlled lab settings. We begin by analyzing the relationship between RSSI (Received Signal Strength Indicator) and packet loss. Next, we benchmark the impact of such losses on the user experience. We conclude evaluating the effectiveness of Sonic pixel interpolation and pushing techniques.

Signal Strength and Sonic Performance. We place five Xiaomi Redmi Go phones, each with the Sonic app installed, at varying distances from a 0.5 W FM transmitter to artificially create diversity in RSSI. All devices are kept fully powered meanwhile 5,000 randomly-selected webpages from the Tranco [146] list are broadcasted over FM at a frequency of 91.5 MHz, so that they are concurrently received by the testing devices while emulating varying RSSI. Figure 2.10 shows the CDF of loss rates across all transmissions and its inset plot presents loss percentage as a function of RSSI range. The main CDF shows that over 95% of transmissions experienced loss rates below 10%. Furthermore, high loss percentages (for the remaining 5% of transmissions) are largely confined to relatively poor signal conditions, particularly in the RSSI range of −70 to −60 dBm. As RSSI improves, the loss percentage rapidly declines and stabilizes near zero.

Pixel Interpolation. [188] evaluates the impact of visual loss and pixel interpolation on perceived content clarity and text readability using feedback from 151 Pakistani university students across 50 test webpages. Their results show that even at a 20% pixel loss rate – which is rare in Sonic as shown in Figure 2.10 – users reported a median content clarity score of 7 out of 10, indicating a generally clear understanding of the page. While text readability was more affected, it remained acceptable at a loss rate of 20%. Rating distributions from this study are provided in Appendix 2.9.

Pushing Metric. We conduct a user study on Prolific [192] to evaluate the likelihood of hyperlink clicks based on visual features of each link. Specifically, we examine the area covered by the link in a webpage screenshot (width · height) and its vertical position on the page (y-position). We randomly sample 100 webpages from the Tranco list. For each page, we generate mobile screenshots along with the bounding-box coordinates (x, y, w, h) of every hyperlink. We then create an interactive webpage where participants are asked to click on the link they would "naturally" choose to visit next.

A total of 100 participants take part in the study, each interacting with 10 different pages. The pages are distributed so that each webpage is evaluated by ten users, resulting in 1,000 total page interactions. Using the data collected from our study, we train a logistic regression model to learn the relative importance of a hyperlink's area and vertical position in predicting click likelihood. The model fits a weighted linear combination of these features to estimate the probability of a link being clicked. The resulting scoring function is:

$$score = 0.68 \cdot w \cdot h - 0.32 \cdot y$$

where `w`, `h`, and `y` denote the width, height, and vertical distance from the top of the page. The negative weight on y-position reflects that links appearing closer to the top (i.e. with lower y-values) are more likely to be clicked. We use this scoring function as our prioritization metric for pushing.

Figure 2.11 shows the likelihood of clicking a hyperlink based on its area and vertical position on the page. The heatmap reveals that links with larger areas and located closer to the top (i.e. lower y-values) are more likely to be clicked. This trend is visible in the gradient transition from red (low click probability) to green (high click probability), moving from the bottom-left to the top-right of the plot. Although the maximum observed click probability is only 0.53—indicating that clicks are far from guaranteed—the relatively higher likelihood still offers a useful signal for prioritization. Since the server remains underutilized during idle periods, pushing these links—even at moderate click probabilities—can improve user experience with minimal additional cost.

2.5 Deployment

This section outlines Sonic deployment at a live FM radio station in Cameroon. We selected this location given its low internet penetration rate comparable to low-income regions (58.1% of Cameroon's population is offline as of 2025 [73]).

Methodology. We start by signing an agreement with an FM radio station in Cameroon to allow the Sonic server to transmit content from 10PM to 5AM daily for six weeks. This overnight window was the only available airtime, as the station's daytime schedule was reserved for regular programming. Such opportunistic use of off-peak radio hours represents the most feasible adoption path for Sonic in the near term. In the future, we envision dedicated FM channels operating full-time for data broadcasting.

We recruited 30 Cameroonians to experiment with Sonic during this period, i.e. request webpages and ask questions to ChatGPT. Study participants were given a Xiaomi Redmi Go phone (featuring Qualcomm Snapdragon 425 processor and 1 GB RAM) flashed with the modified version of LineageOS, and Sonic app pre-installed. To send requests, each phone had a SIM card with an unlimited SMS bundle.

As shown in Figure 2.12, the Sonic server was set up at the FM radio station using a MacBook Air with 8 GB of RAM, running the Sonic Docker container. We used Huawei's E8372h-320 LTE/4G USB Mobile WiFi Dongle [114] to interface with the SIM card via `huawei-lte-api` [215] and receive incoming SMS messages. The Sonic app was programmed to send SMS messages to the number associated with the SIM card used by the dongle. For internet access, we used a mobile data subscription from MTN Cameroon (see Section 2.2 for details on plan and connection quality).

Data Collection. Table 2.1 outlines our deployment. A total of 30 participants were recruited, divided into two sequential batches of 15 participants each. The study spanned 6 weeks in total, with each batch participating for 3 weeks. As an incentive to the participants, we offered USD 2 per person per day. Before the study, participants signed a consent form and were allowed to withdraw from the study at any time. The participants were then given an introduction on how to use the Sonic app; further, an institutional review board (IRB) approval was granted to conduct the study. The authors who conducted the study are CITI [63] certified. No sensitive or personal information of the participants was collected, except for their name and phone number to contact

Property	Description
Location	Cameroon
Participants	30
Number of batches	2
Participants per batch	15
Duration of study	6 weeks
Duration per batch	3 weeks
Transmission window	10 PM to 5 AM daily
Daily request quota	10 (GPT + URL)
Total URL requests	1,737
Total GPT requests	2,936
Median requests per user	Total: 160, GPT: 96, URL: 64 (in 3 weeks)

Table 2.1: Summary of Sonic deployment.

them and disburse the incentive money at the end of the experiment.

Study participants were allowed to make up to 10 requests per day–this included both webpage URL requests and GPT queries. Participants were allowed to make requests at anytime during the day; however, responses were transmitted during the transmission window of 7 hours (10 PM to 5 AM). Over 3 weeks, participants made 1,737 URL requests and 2,936 GPT queries in total. The median number of requests per user was 160, with 96 GPT queries and 64 URL requests.

Deployment Challenges. We encountered several challenges during Sonic deployment. Initially, airport security confiscated 10 mobile phones intended for participants, significantly reducing the number of devices available for deployment. Only 15 phones ultimately reached Cameroon, forcing us to conduct the experiment in two separate batches. Securing reliable internet connectivity for the Sonic server also proved challenging: even the best available mobile internet plan from MTN was unstable (as discussed in Section 2.2), occasionally unavailable for entire days, and affected by significant latency. Compounding these issues, the village where we deployed frequently experienced electricity outages—lasting up to 8 hours and often overlapping with the transmission window—completely disrupting FM radio transmissions. During these outages, although users continued to request content via SMS, they were unable to receive any responses.

Operational issues further complicated the deployment. In the first batch, the Sonic app sent acknowledgment messages (ACKs) for all received transmissions, enabled by unlimited SMS bundles purchased for each participant. However, the resulting high volume of SMS traffic quickly raised suspicion with the mobile operator, leading to the blocking of all deployed SIM cards. Consequently, we had to disable the ACK mechanism, leaving us without real-time operational feedback or heartbeat signals from participants' phones. Additionally, the success of FM radio transmissions critically depended on the radio station staff accurately switching to Sonic broadcasts at exactly 10 PM each night. This switch-over, however, was inconsistent, resulting in multiple days without any transmissions. We also discovered that the transmitter's output volume needed to be set to 100% to achieve better range and improve transmission quality, but maintaining this setting consistently proved challenging for the radio staff as well. Finally, the effectiveness of the Sonic system relied heavily on participants regularly charging their phones and keeping earphones connected at all times, as the earphones served as antennas for receiving broadcasts. Collectively, these logistical and operational hurdles made it difficult to maintain ideal conditions for Sonic deployment.

2.6 Results

In this section, we present our analysis from Sonic deployment at a live FM radio station in Cameroon.

RSSI and Loss Analysis. Figure 2.13 shows the spatial distribution of RSSI measurements across a 100m-resolution grid around the FM radio station. Using ordinary kriging [246], we interpolate user-collected GPS-tagged signal data to generate a continuous RSSI map. While the radio station is centrally located, the strongest signal regions are notably offset to the northeast, with two additional users registering high RSSI values at distances of approximately 900–950 m (one directly to the south and another to the southwest). Contrary to the expected radial decay in signal strength with distance, these observations demonstrate the influence of antenna height and placement similar to what was observed by [32]. The northeastward bias in signal strength likely results from the antenna's physical orientation or directional configuration, while the isolated strong-signal detections at longer distances suggest favorable line-of-sight conditions.

Figure 2.14 shows transmission loss percentages across RSSI ranges, with each violin's width representing the proportion of total transmissions in that range (e.g. 22.1% of all transmissions occurred between -70 and -60 dBm). We observe a clear trend: better signal strength (i.e. higher RSSI) is associated with reduced transmission loss. In poor signal conditions below -90 dBm, the loss rate is frequently near 100%. In contrast, for stronger signals in the -80 to -50 dBm range (accounting for above 60% of total transmissions), data points are more concentrated below 20%, and instances of 100% loss are rare. This confirms, in the wild, the observations from our benchmarking under controlled settings (see Figure 2.10).

To investigate per user performance, Figure 2.17 shows boxplots of RSSI values per participant alongside their completion rates for both GPT and URL requests. Users with stronger median RSSI values (e.g. above −70 dBm) consistently achieve high completion rates across both content types. For instance, users 15–17, 21–28, and 30 report more than 80% completion rates, indicating that strong and stable signal conditions are sufficient for reliable content delivery. In contrast, users with lower and unstable signal quality, particularly those with median RSSI fluctuating below −90 dBm (e.g. users 4, 12, and 19), show drastically reduced completion rates, often below 30%.

Due to their smaller sizes, GPT responses have shorter broadcast duration than webpages. However, they are more susceptible to failures from partial frame loss. A single 500-byte frame drop can disrupt an entire GPT message, whereas similar loss in a webpage screenshot has limited impact (see Figure 2.20). As a result, webpages achieve higher completion rates. This can be observed for users 7, 8, 10, and 11, where URL completions are consistently higher than GPT under similar signal conditions. This trend is also reflected in Figure 2.15, which shows the CDF of loss rates observed for the two types of content—GPT responses and URL transmissions. URL transmissions (orange line) appear more loss-resilient than GPT (blue line) due to their shorter duration and stricter tolerance to partial frame loss.

Scalability. Figure 2.16 shows the evolution of the transmission queue over the course of a day. We select the busiest day to understand transmission trends within the transmission window (marked by the vertical dashed lines at 22:00 and 05:00). The blue trace represents the real deployment of Sonic, with 15 users receiving content on a single frequency. Although each user requested 10 pages, on average, the queue size peaked at 103 items, indicating that roughly 30% of transmissions were served from the cache. The sharp spike at 09:30 corresponds to a scheduled "push" of pre-selected news pages during an otherwise idle period. All other curves represent FCFS (First Come First Serve) queue simulations, scaling this baseline traffic to heavier loads. For example, with 30 users on a single FM frequency (orange), the queue peaks at around 200 items but is still fully transmitted within the 7-hour transmission window. This suggests that one frequency can support up to 30 active users with similar queuing patterns. When capacity increases to two frequencies and 105 users (green), the peak backlog rises to 800 items, half of which could not be transmitted before 05:00. Similar patterns are observed for 150 users with three and four available frequencies, respectively. However, a configuration of 300 users with 10 available frequencies appears sufficient to fully transmit the queue within the transmission window.

Content Analysis. Figure 2.18 presents a treemap of the top 10 content categories for both GPT queries and URL requests made by Sonic users. GPT queries are classified using Meta's Llama 3 model [95], while domains in URL requests are categorized using Cloudflare's Domain Intelligence API [65] following the methodology in [212]. Among GPT queries, Geography dominates with 22% of all requests, followed by Politics (9%), Sports (6%), History (5%), Technology (4%), and Medicine (4%). Smaller but still notable portions of queries are related to Philosophy, Science, Business, and Chemistry. For URL requests, News & Media (12%) and Business (10%) are the most common, followed by Technology and Education. Users also accessed sites in categories such as Search Engines, Ecommerce, Travel, and Video Streaming. However, content from these latter categories is less likely to be useful given Sonic's current limitation of only supporting page screenshots with limited (and slow) hyperlink interactivity.

User Experience. At the end of the experiment, we asked participants to complete an exit survey assessing their experience with Sonic across three dimensions. Each question was rated on a 5-point likert scale, with varying response ranges depending on the aspect evaluated. For system reliability, users responded to: *"How reliable was the content received through the Sonic app?"* (1 = Not reliable at all, 5 = Very reliable). For UI intuitiveness, they answered: *"How intuitive is the user interface of the Sonic app?"* (1 = Not intuitive at all, 5 = Very intuitive). Lastly, for content relevance, users rated: *"How useful was the content you received from the system?"* (1 = Not useful at all, 5 = Very useful).

Figure 2.19 shows the CDF of responses to the survey questions. Overall, user feedback was positive: 72% of participants rated the Sonic app as "intuitive" and the content as "useful" (with scores of 4 or 5). In contrast, system reliability received relatively lower ratings, with only 62% of users selecting 4 or 5. This is expected, as some participants experienced fluctuating or low RSSI values, leading to reduced request completion rates, as discussed earlier in Figure 2.17.

2.7 Related Work

Improving internet accessibility in developing regions is crucial for delivering essential services such as education and healthcare. However, challenges like unreliable hardware, limited cellular coverage, high data costs, and increasingly complex webpages hinder connectivity. Existing efforts focus on web simplification and new access technologies tailored for these regions. Sonic combines *web simplification* with *data-over-sound* transmission to enable connectivity in rural and remote areas. In the following, we discuss related works in both research areas.

Web Simplification. Prior work has explored reducing webpage complexity to improve performance under limited connectivity. Habib et al. [101] proposed a framework that dynamically adapts webpage complexity based on network conditions. Muzeel [142] removes unused JavaScript, while MAML [185, 186] introduces a minimalist specification language omitting JavaScript and CSS. Klotski [45] prioritizes user-relevant content, Shandian [249] restructures loading via split-browser design, and Polaris [176] accelerates rendering using dependency graphs. Though effective under constrained bandwidth, these methods require basic internet access, which is unavailable or unaffordable in many rural areas.

Data over FM. To our knowledge, Sonic is the first system to leverage FM radio as a means to broadcast internet connectivity in rural regions. The process of encoding webpages as sound is inspired by several research papers [31, 132, 147, 210, 217] and open source tools [29, 89, 194] that have explored how to transmit data over sound at inaudible audio frequencies, i.e. above 18kHz. The usage of FM radio for novel applications has also been explored by in previous works including RevCast [218] which leverages the broadcast nature of FM radio for certificate revocation.[38, 39] use FM radio broadcasting to disseminate warning information to drivers. In 2003, Microsoft used FM subcarrier signals to turn ordinary gadgets into smart gadgets. MSN Direct [170] was a subscription network which sent short text updates over DirectBand, a 67.65 kHz subcarrier leased by Microsoft from commercial radio broadcasters.

2.8 Conclusion

Despite decades of progress in global connectivity, billions of people around the world remain offline—not due to a lack of infrastructure, but because of persistent affordability barriers. Access to the internet continues to be out of reach for many, especially in resource-constrained regions where even low-cost mobile data can be prohibitively expensive. In this context, Sonic introduces a novel, ultra-low-cost approach to narrowing the digital divide by leveraging FM radio—a ubiquitous, inexpensive, and underutilized medium—to deliver essential web content and large language model (LLM)-based interactions without requiring an internet connection. By combining a full-system design, seamless integration with Android FM tuners, and deployment in real-world settings such as Cameroon, we demonstrate that Sonic can reliably transmit simplified web content and AI-generated responses in a way that is accessible, scalable, and resilient. Our work showcases the untapped potential of repurposing existing broadcast infrastructure to extend digital access to underserved populations, offering a practical path forward for connecting the unconnected and promoting more equitable access to knowledge and services across the globe.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'M'      |      'D'      |      'T'      |      'A'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type (3 bytes, eg: "img", "url")       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|               Partition Sizes CSV                             |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'U'      |      'R'      |      'L'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Request URL                                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'I'      |      'D'      |      'E'      |      'N'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Response Identifier (8 bytes)                   |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'W'      |       Page Width (3 bytes)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'H'      |       Page Height (3 bytes)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'E'      |      'O'      |      'M'      |      'D'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'L'      |      'N'      |      'K'      |      'S'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|               JSON Array [{x, y, w, h, href}]                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'S'      |      'D'      |      'T'      |      'A'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 500-byte Frame Data Array:                                    |
|      'C'      |      '1'      |      '3'      |      '7'      |
| Frame Index + Partition Index + Data                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'I'      |      'D'      |      'E'      |      'N'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Response Identifier (8 bytes)                   |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      'E'      |      'O'      |      'F'      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.7: Sonic file format.

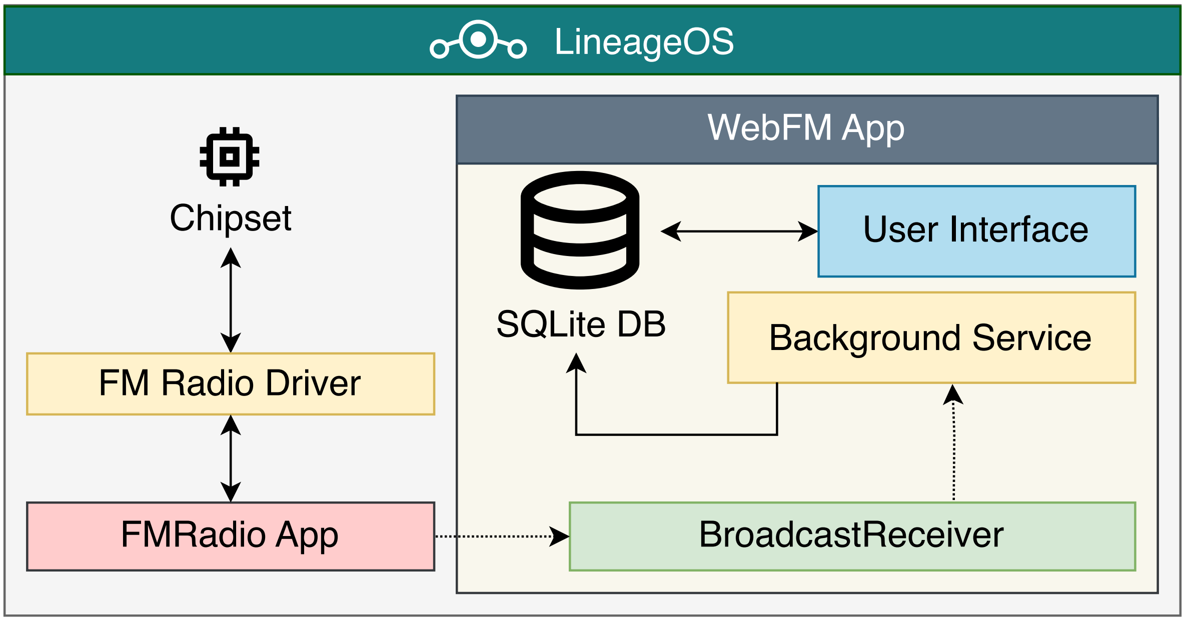
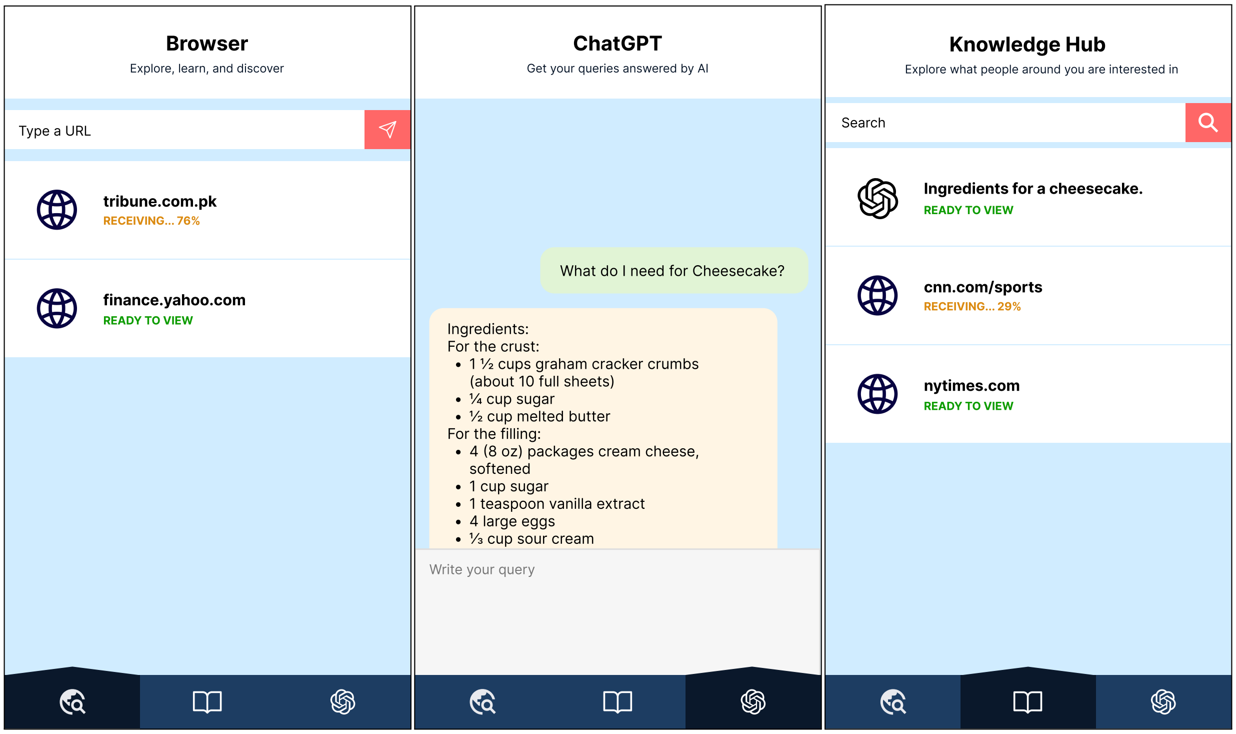


Figure 2.8: Sonic client architecture.



What do I need for Cheesecake?

Ingredients:
For the crust:
- 1 ½ cups graham cracker crumbs (about 10 full sheets)
- ¼ cup sugar
- ½ cup melted butter

For the filling:
- 4 (8 oz) packages cream cheese, softened
- 1 cup sugar
- 1 teaspoon vanilla extract
- 4 large eggs
- ⅓ cup sour cream

Figure 2.9: Sonic's user interface.

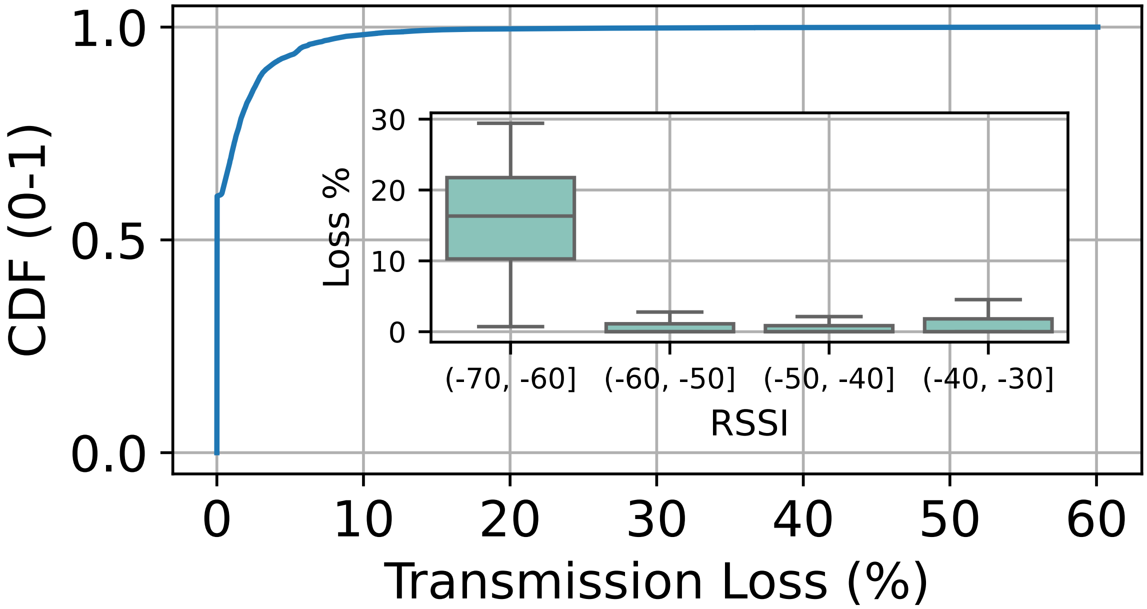


Figure 2.10: CDF of loss percentage under ideal conditions. Inset shows loss % vs RSSI.

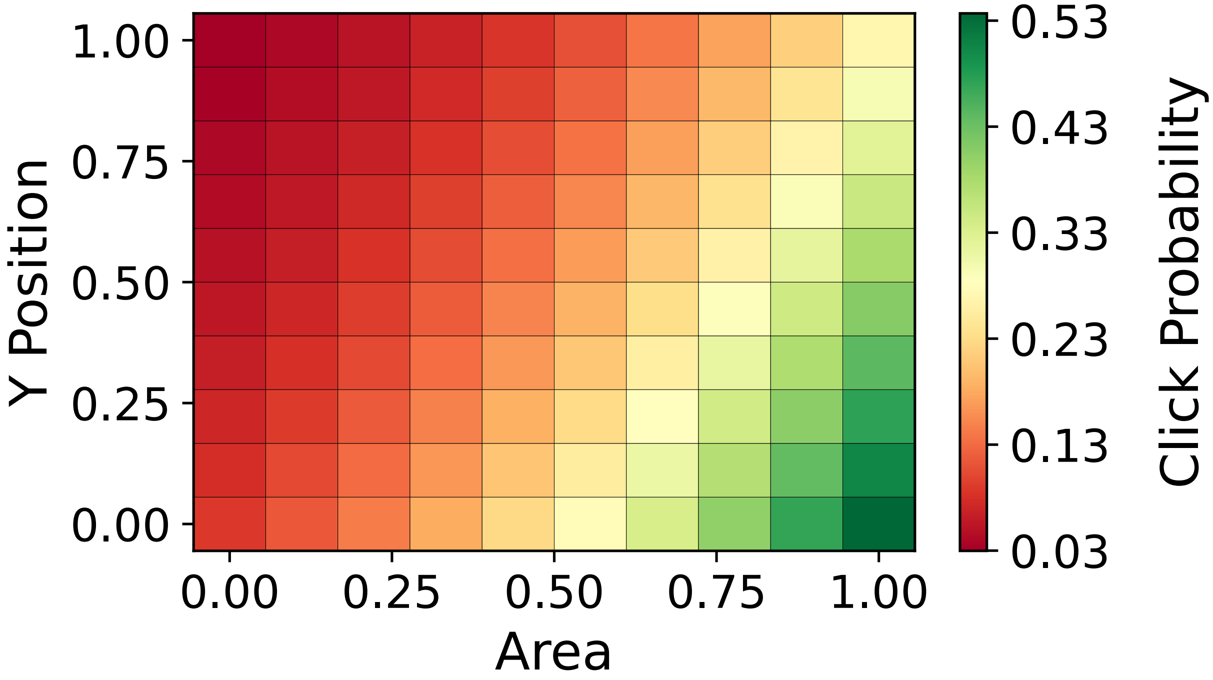


Figure 2.11: Likelihood of clicking on a hyperlink in a webpage given its area and y position. Data from 1,000 page interactions from 100 users across 100 pages.

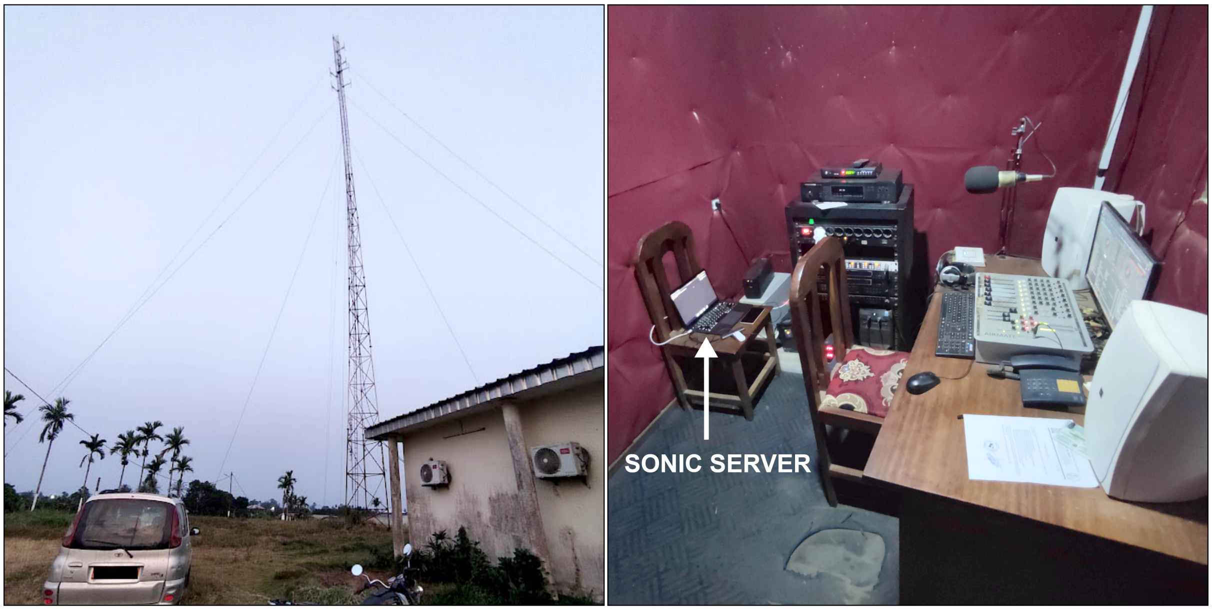


Figure 2.12: FM radio station in Cameroon, and Sonic server located inside the station.

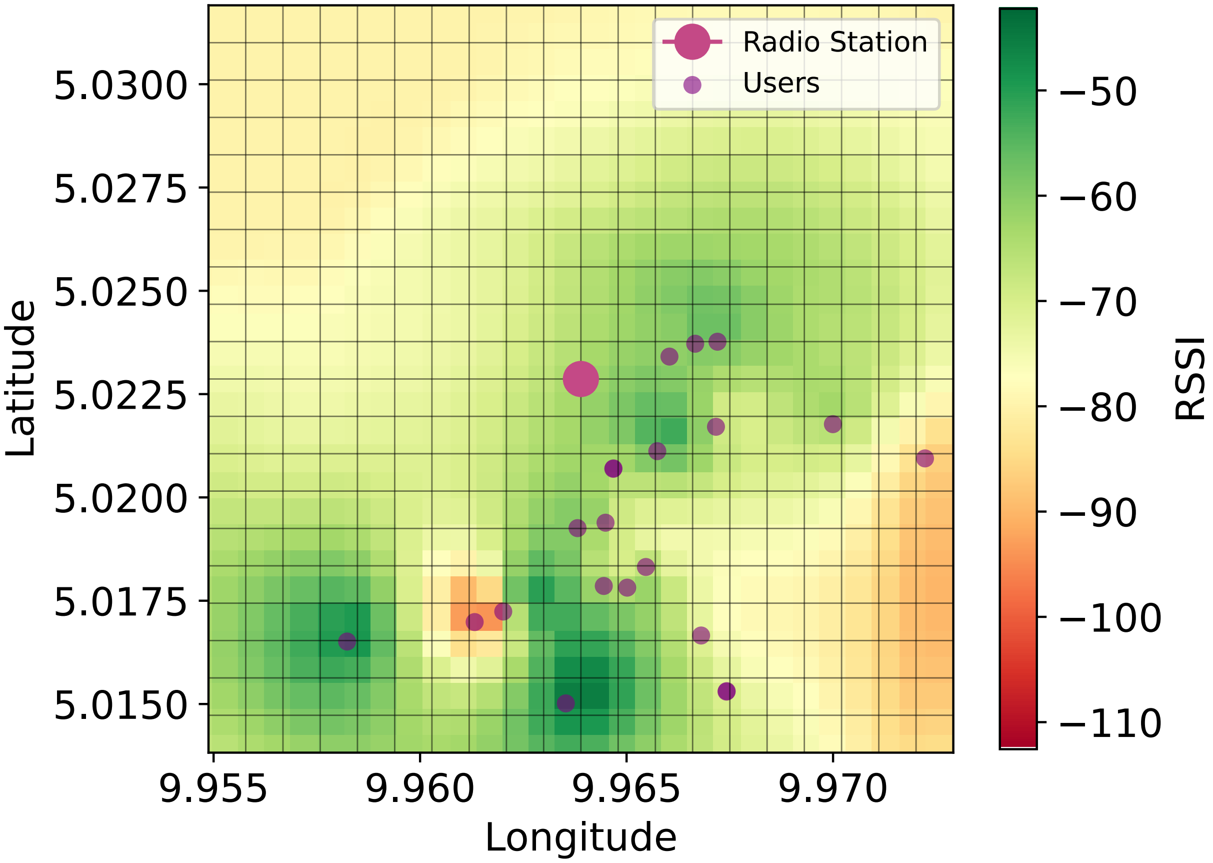


Figure 2.13: Location map of the 30 Sonic users with their interpolated RSSI values. Each cell in the grid is 100 meters in scale. Note that some users are overlapping.

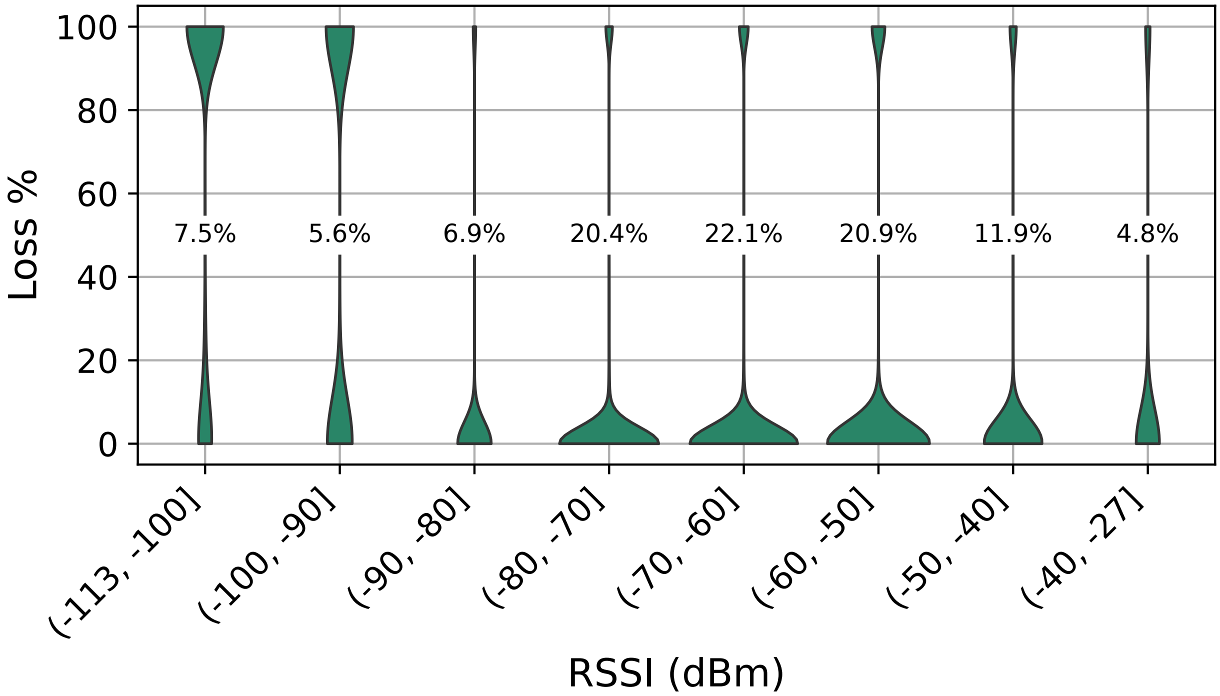


Figure 2.14: Loss % as a function of RSSI range. Violin's width is proportional to the percentage of total transmissions occurred in the RSSI range.

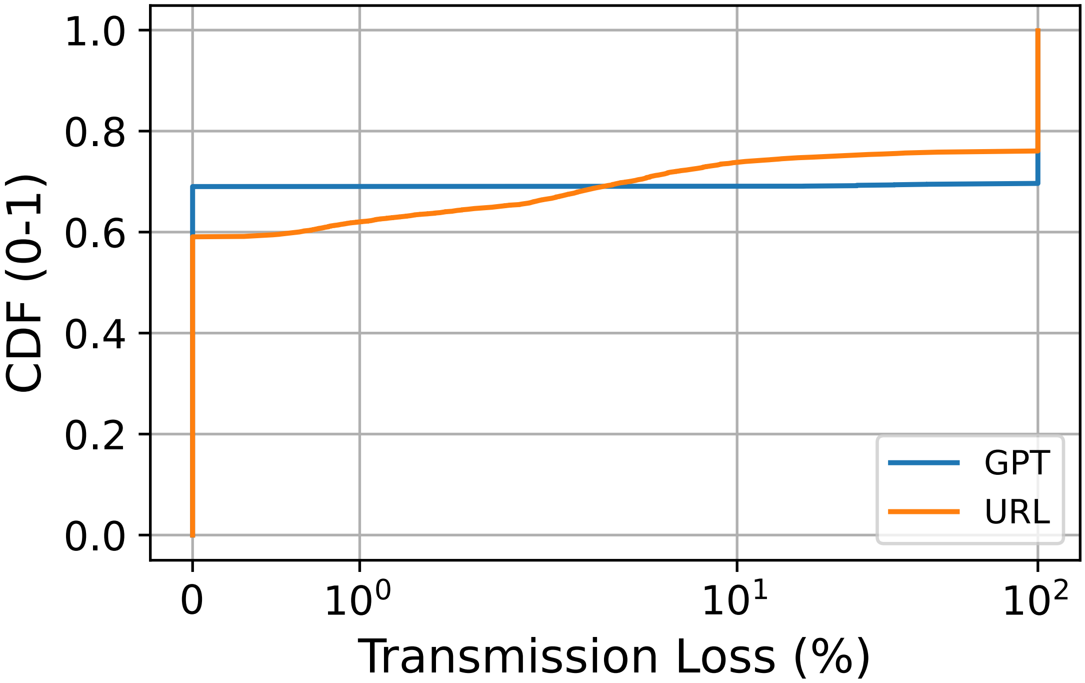


Figure 2.15: CDF of loss % for GPT and URL transmissions.

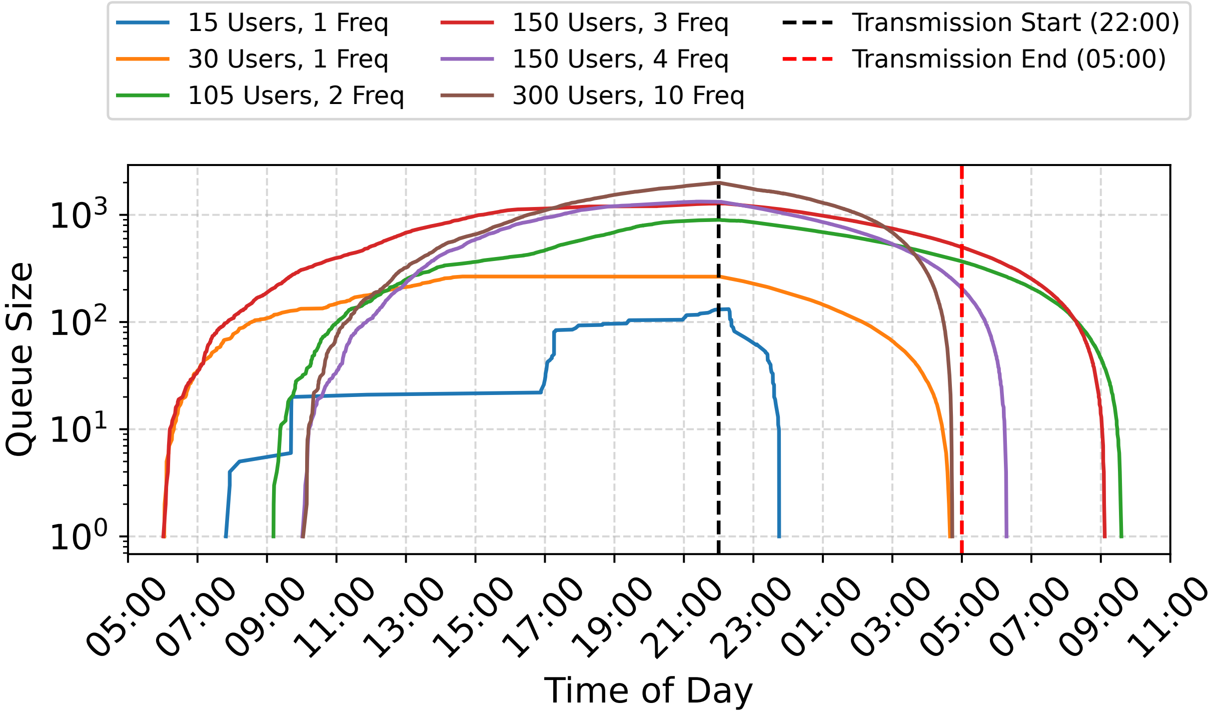


Figure 2.16: Queue size for *actual* (15 users, 1 frequency), and *simulated* (30-300 users, 1-10 frequencies) loads assuming the busiest day of Sonic deployment.

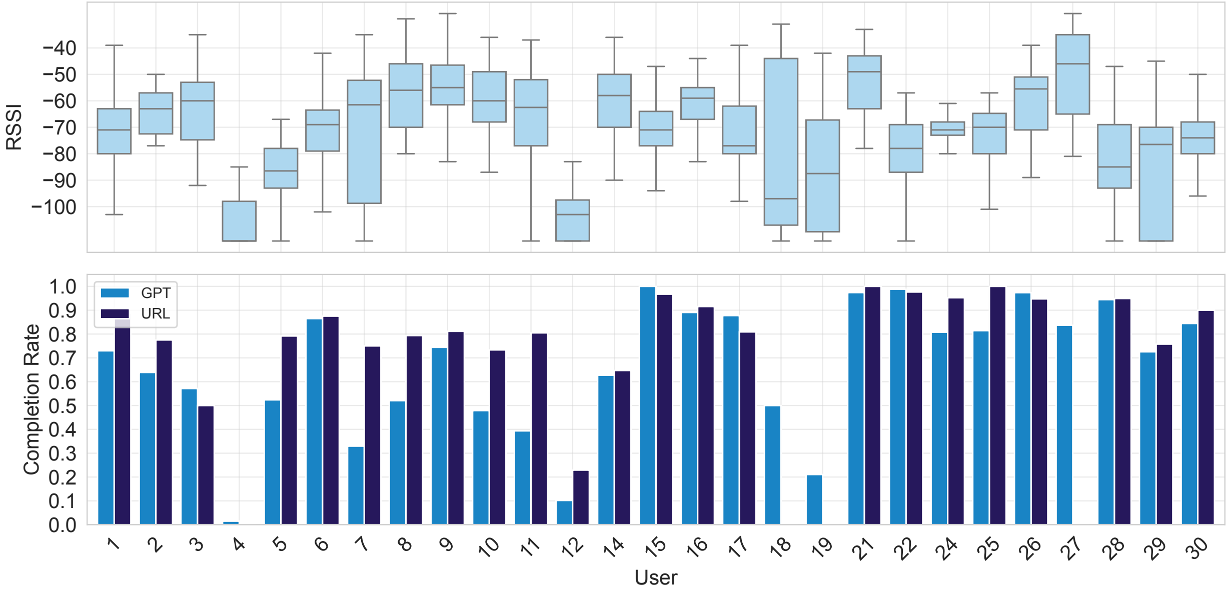


Figure 2.17: RSSI distribution per user with their request completion rates. Users 13, 20, and 23 were removed as their devices appeared faulty; they neither recorded any RSSI measurements nor received any transmissions.

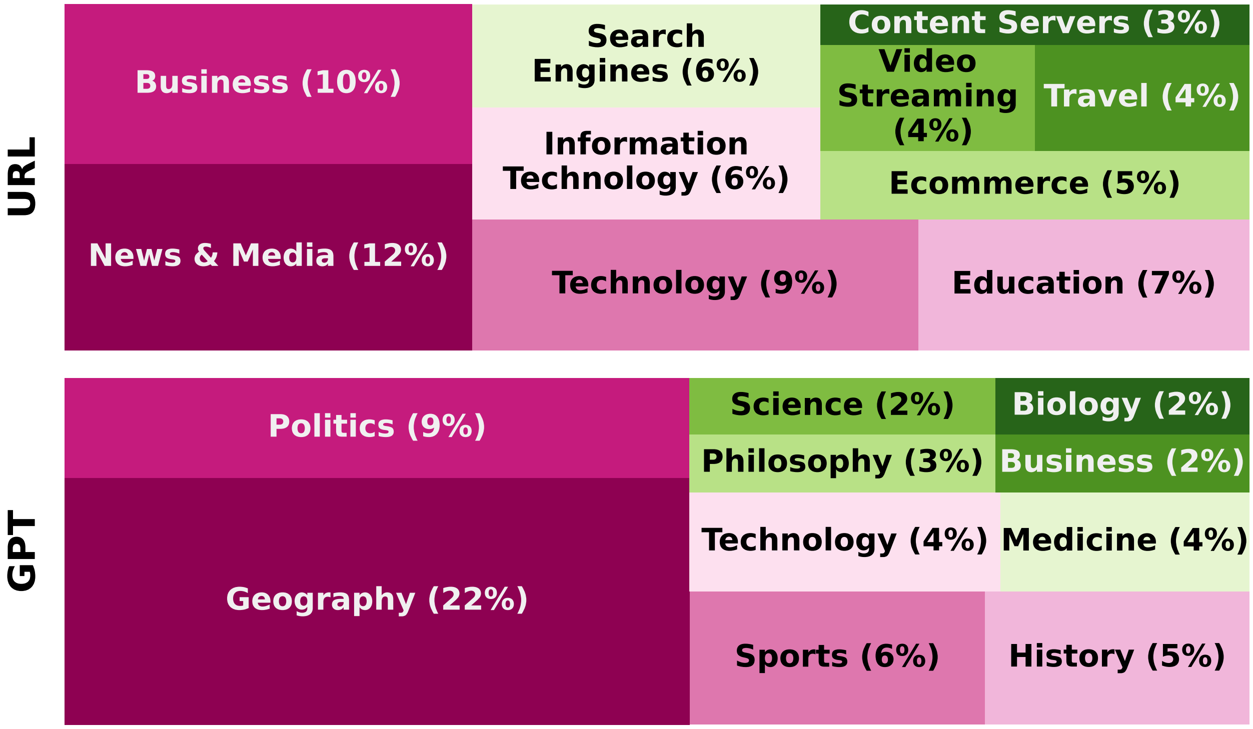


Figure 2.18: Top 10 categories of URL and GPT content requested by Sonic users.

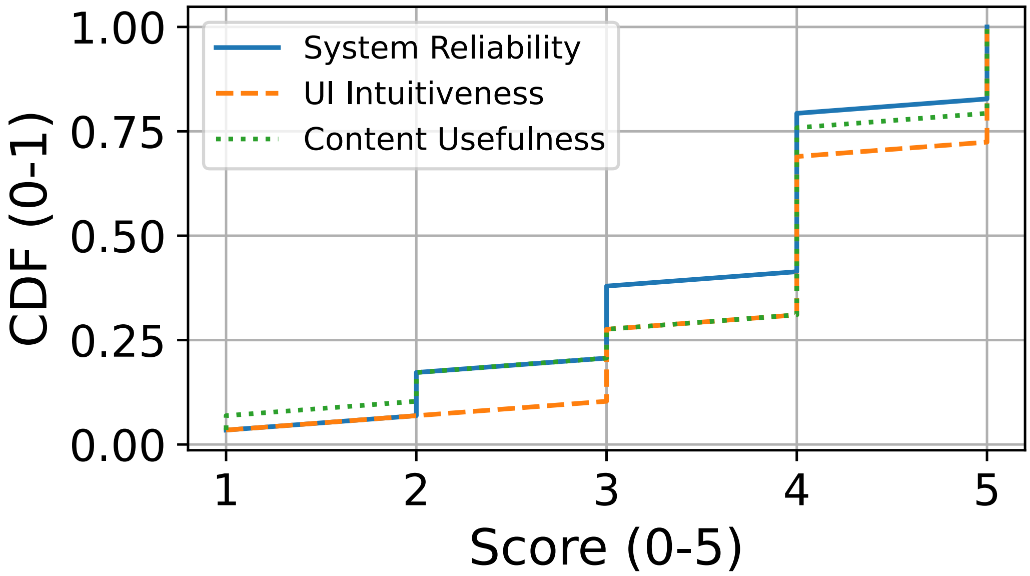


Figure 2.19: User experience of Sonic on system reliability, UI intuitiveness, and content usefulness.

2.9 Pixel Interpolation

Figure 2.20 contains the results from a user study conducted by [188] to benchmark the effectiveness of their pixel interpolation approach. The study simulates varying levels of visual loss on popular Pakistani webpages and evaluates their impact on perceived readability and content clarity. Screenshots of the top 50 webpages from the Tranco list were captured and processed under four levels of synthetic visual loss (5%, 10%, 20%, and 50%). Each screenshot was rendered in two variants—one with missing pixels left dark, and another corrected using nearest-neighbor pixel interpolation (detailed in Section 2.3.2)—yielding 400 total images.

A total of 151 university students in Pakistan participated in the study, each rating 20 randomly assigned screenshots such that each image received at least seven responses. Ratings were collected on a 0–10 Likert scale across two dimensions: (a) content clarity (*How well can you understand the content in this image?*), and (b) text readability (*How readable is the text in the image given the noise?*). Figure 2.20 presents median boxplots per webpage, with hatched bars representing content clarity and plain bars representing text readability. Even with a 20% pixel loss rate, participants reported a median content clarity score of 7 out of 10, suggesting that the overall understanding of the page remained largely intact. Although text readability was more impacted, it continued to be within an acceptable range at 20% loss.

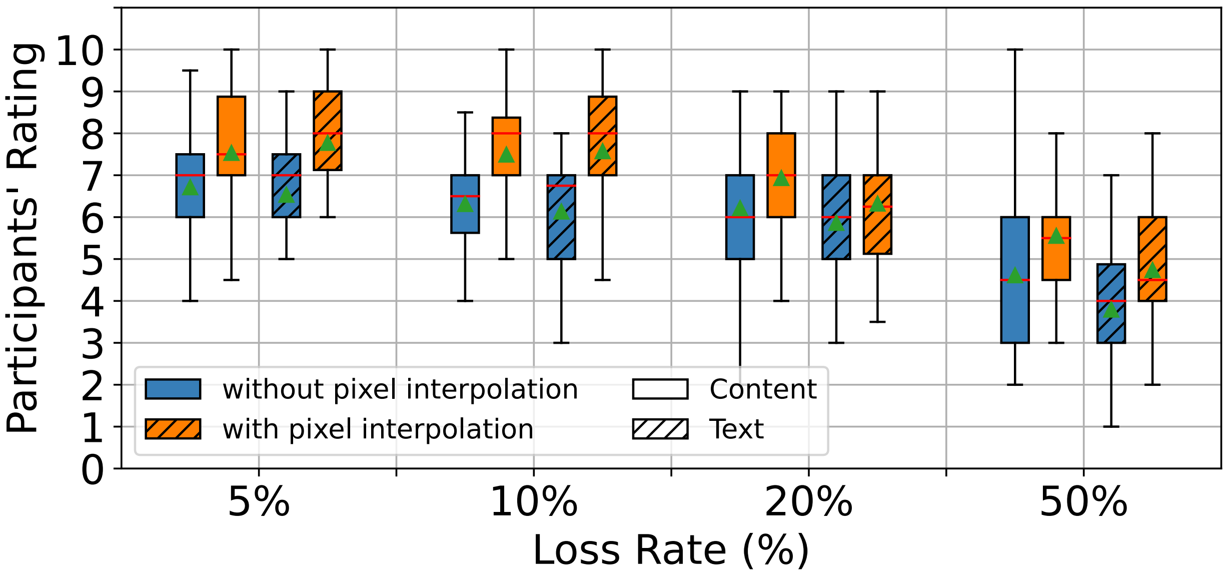


Figure 2.20: (b) Distribution of user ratings (0–10) for top 50 Pakistani webpages, with/without interpolation.

3 | Towards a Web Without Digital Inequality

This chapter is adapted from "Towards a World Wide Web without digital inequality" published in the Proceedings of the National Academy of Sciences (PNAS) [59]. In this chapter, we discuss Lite-Web, a hybrid solution to imrpove web accessibility in emerging regions with limited network infrastructure.

Abstract

The World Wide Web empowers people in developing regions by eradicating illiteracy, supporting women, and generating economic opportunities. However, their reliance on limited bandwidth and low-end phones leaves them with a poorer browsing experience compared to privileged users across the digital divide. To evaluate the extent of this phenomenon, we sent participants to 56 cities to measure the cost of mobile data and the average page load time. We found the cost to be orders of magnitude greater, and the average page load time to be four times slower, in some locations compared to others. Analyzing how popular webpages have changed over the past years suggests that they are increasingly designed with high processing power in mind, effectively leaving the less fortunate users behind. Addressing this digital inequality through new infrastructure takes years to complete and billions of dollars to finance. A more practical solution

is to make the webpages more accessible by reducing their size and optimizing their load time. To this end, we developed a solution called Lite-Web, and evaluated it in the Gilgit-Baltistan province of Pakistan, demonstrating that it transforms the browsing experience of a Pakistani villager using a low-end phone to almost that of a Dubai resident using a flagship phone. A user study in two high schools in Pakistan confirms that the performance gains come at no expense to the pages' look and functionality. These findings suggest that deploying Lite-Web at scale would constitute a major step towards a World Wide Web without digital inequality.

3.1 Introduction

The World Wide Web (WWW) was envisioned as an egalitarian platform that provides universal access to the wealth of accumulated human knowledge. It has enabled the creation of projects such as Wikipedia, Khan Academy, and Massive Open Online Courses (MOOCs), all of which hold the promise of democratizing education [105]. In developing regions, the WWW has contributed to women's empowerment by offering a gender-opaque medium that alleviates bias, provides access to distance learning and employment opportunities, and increases the chances of receiving support from organizations concerned with the well-being of women [119, 258]. Another way in which the WWW supports the developing regions is by generating economic opportunities. For example, it has been shown that fast Internet access can decrease (un)employment inequality in Africa [112], and mobile broadband access can decrease poverty, particularly among rural households [30]. Additionally, the development of e-commerce can play a significant role in narrowing the urban–rural income gap in China [149]. Access to critical information empowers farmers and fishermen in emerging markets. For example, by tracking weather conditions and comparing wholesale prices, farmers and fishermen in India increased their profit by 8%, eventually leading to a 4% decrease in prices for their customers [128]. The WWW is even helping eradicate illiteracy—one of the main barriers to digital inclusion. For example, in Sub-Saharan

Africa, where most people do not own any books, the massive proliferation of mobile devices allows people to develop, sustain and enhance their literacy skills by providing a medium through which they can access reading materials [254]. Moreover, providing access to online material in Malawian boarding schools can encourage reading and improve educational outcomes [75]. Other examples include non-profit initiatives such as `remoteStudentExchange.org`, which in the few months since its launch in January 2021 has given thousands of students in low- and middle-income countries direct access to (online) courses from the world's leading universities.

'The growing adoption of mobile phones has contributed to the significant increase in Internet access over the past decade. In 2018, nearly 300 million users were newly connected to the mobile web, and in 2019, the total number of mobile web users exceeded 3.5 billion worldwide. Of those users, 74% live in low- and middle-income countries [97], where mobile phones are the primary means of Internet access. Many of those users depend solely on mobile phones. For instance, across 18 developing countries, an average of 57% of Internet access in 2018 was carried out exclusively via mobile phones [97]. A key enabler of mobile Internet adoption is affordability; not only is mobile data becoming available at lower prices [157], but also mobile phones are becoming more affordable. For instance, cheaper phones are expected to be available in Pakistan [124], India [130], and Africa [222] in the near future, as a new generation of phones is expected to be made available for only $20 [5]. Although access to the mobile web is expanding in developing countries [236] to the point of surpassing access to piped water and consistent electricity [168], the user experience remains poor [228, 267]. This is part of a larger phenomenon known as the *digital divide*, which separates those with high-quality access to information and communications technologies from those with poorer alternatives [236]. Our primary goal is to evaluate the extent of this phenomenon worldwide, and to explore affordable and scalable solutions that can potentially bridge the divide and alleviate the digital inequality experienced by underserved communities.

3.2 Results

Understanding digital inequality

To better understand the variation in web access quality across the globe, we needed to send participants to different cities spanning six continents, and have each of them access the same set of webpages (to control the browsing experience) using the exact same hardware (to control the processing power) and the same web browser (Google Chrome in our case) at the same local time (12:00 pm in our case) while being connected to a cellular network (rather than Wi-Fi) to ensure that any observed differences in average page load time are not influenced by variations in these factors. To this end, we leveraged the diversity of the student population at New York University Abu Dhabi by recruiting undergraduates traveling back to their home countries during the winter break. Each participant was handed the same low-end phone model, namely Xiaomi Redmi Go, and was asked to install a tool on their laptop, called *WebPageTest* [253], which automates web requests on that phone while recording various page load time metrics using the actual (rather than emulated) connection speed. Those web requests were for the 100 webpages that were most frequently visited worldwide at the time according to Alexa [135].

The students that we recruited ended up visiting 72 cities across six continents. Upon their arrival at their respective destinations, participants purchased a SIM card along with an affordable pricing plan from a local service provider, and kept the receipt which specified the total cost and the number of Gigabytes provided by the plan. After that, they connected the phone to their laptop via a USB cable, and ran the tool at 12:00 pm local time to automatically request the 100 webpages via Google Chrome on the phone and extract the results. The experiment took place in December 2019 and January 2020. Students who failed to follow the experimental protocol were discarded from our analysis, yielding a total of 56 cities.

When comparing the price of 1 Gigabyte across countries, one needs to take into consideration the differences in living standards. For instance, even if the cost of 1 Gigabyte cost in a rich country was the same as that in a poor country (e.g., 1 USD) this may be considered affordable in the former but not the latter, e.g., due to differences in average salaries. Thus, following common practice in economics [139], we use the purchasing power parity (PPP) in each country as an exchange rate to convert the value of 1 Gigabyte in their local currency to their equivalent value in USD, thereby reflecting the difference in the standard of living between countries. Fig. 3.1a summarizes the results of our experiment, where circles correspond to locations, colors represent average page load time, and diameters represent adjusted costs per Gigabyte. As can be seen, there is a clear digital inequality across the globe. The average page load time in some locations is four times longer than in others (about 47 vs. 12 seconds), and the cost per Gigabyte is orders of magnitude greater than in others ($43 vs. $0.08); see Supplementary Table 1 for numeric values. Similar results were obtained when using direct conversion rates (Supplementary Fig. 1) and when using gross domestic product (GDP) in purchasing power parity (PPP) for each country (Supplementary Fig. 2).

To facilitate the comparison between the different locations, we plotted the distribution of the cost of one Gigabyte per location (Fig. 3.1b) as well as the distribution of the page load time per location, averaged over different webpages (Fig. 3.1c). Indeed, these distributions highlight the inequality between the locations. Moreover, to understand how the webpages themselves differ in terms of their complexity, we plotted the distribution of the page load time (seconds) per webpage, averaged over different locations (Fig. 3.1d). As can be seen, the page load time differs greatly across the webpages, ranging from 3.6 to 62.6, with the mean being 20.8 (note that this is the time required to load the entire page).

Since the hardware specifications can affect the page load time, all measurements were taken using the same phone model, ensuring that the specifications were unified across locations and webpages. A low-end phone—Xiaomi Redmi Go—was used to help us understand the web browsing experience of disadvantaged users; see Supplementary Table 2 for the technical specifications of this phone. Note, however, that users with high income may afford high-end phones instead, which would make the inequality even greater. We found that the page load time and the cost per Gigabyte are not related to population size. Furthermore, when comparing capital to non-capital cities, we found the page load time to be almost identical, and the cost per Gigabyte to be twice as high in capital cities. Interestingly, we found a positive correlation ($r = 0.46$, $p = 0.0004$, Supplementary Fig. 3) between page load time and cost per Gigabyte, indicating that those with poorer connection quality pay more, not less, than their counterparts.

JavaScript impact on digital inequality

Arguably, digital inequality can be eliminated by providing cheap, fast connections worldwide. Unfortunately, this would not only take years to accomplish, but would also be extremely costly, e.g., achieving universal, affordable, and good quality Internet access in Africa by 2030 would require 100 billion US dollars [66]. A significantly cheaper alternative would be to make the webpages themselves "lighter", by reducing their bandwidth and processing requirements. Such a solution would be desirable even if the lighter versions were slightly different from the original pages, as long as the compromise to the user experience is minimal. However, given the myriad webpages in the WWW, it may seem infeasible to analyze them all to identify the elements that are costly (in terms of bandwidth and processing time) and non-essential to the webpage (in terms of appearance and functionality).

Our key insight is to focus on JavaScript elements, which are not only computationally intensive, but are also widely-used across the WWW [82]. Processing these elements is more demanding for web browsers than equivalently-sized web components [184]. Moreover, the download size of these elements often represents a considerable percentage of the total download size per page [201, 202]. Surprisingly, despite its ubiquity, the cost of JavaScript processing on page load time is not fully understood to date. Motivated by this observation, we went six years back in time to understand how the processing of JavaScript affected the web browsing experience on

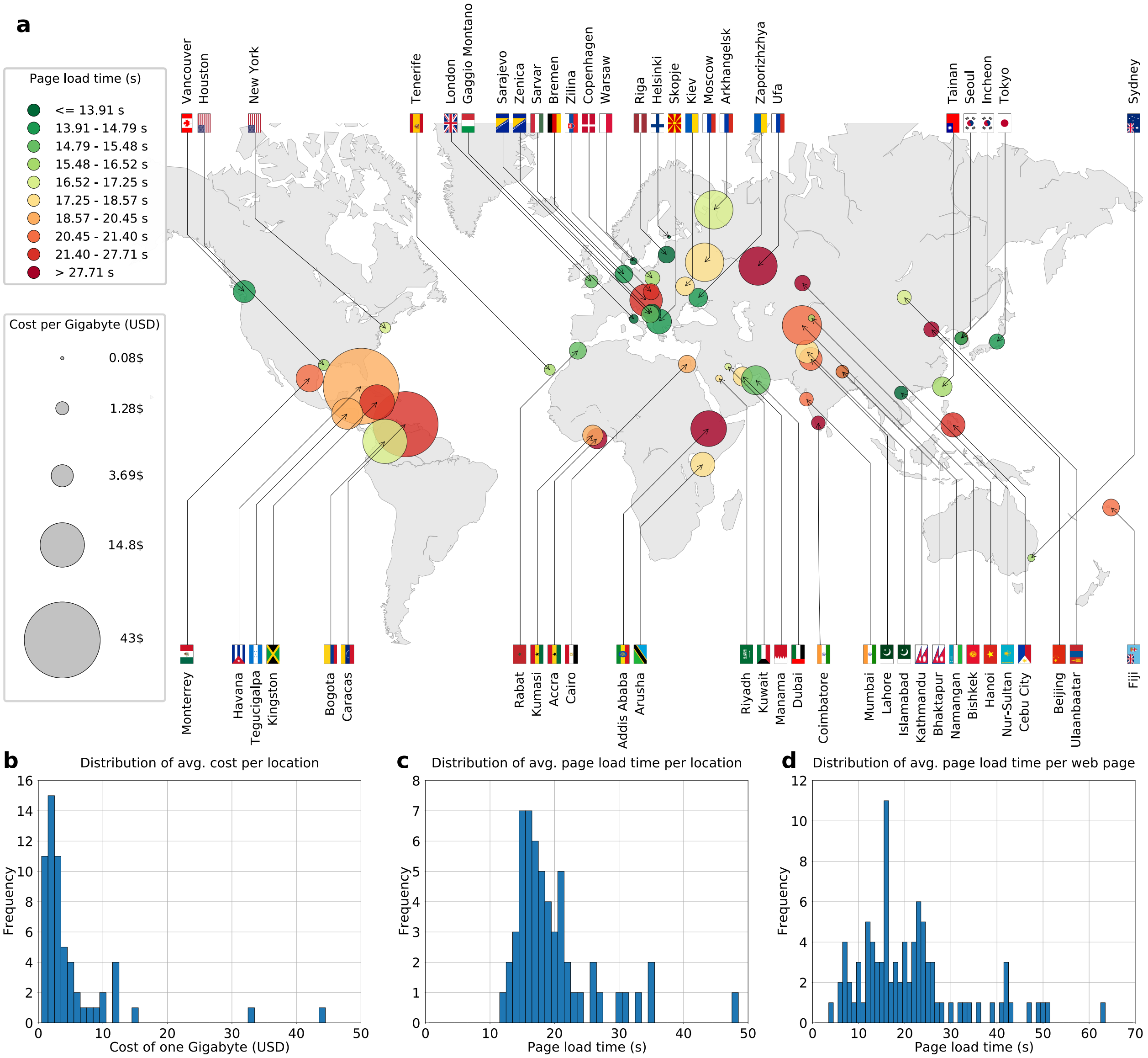


Figure 3.1: Average page load time and data cost across different locations. **a**, Each location is represented by a circle whose diameter reflects the costs per Gigabyte (measured based on the purchasing power parity), and whose color represents the average page load time (measured in seconds) of the 100 most frequently visited pages worldwide. Page load times were measured by accessing the pages via the same low-end mobile phone model—Xiaomi Redmi Go—using a cellular network at that location. **b**, Distribution of the cost of 1 Gigabyte (USD) per location. **c**, Distribution of the page load time (seconds) per location, averaged over different webpages. **d**, Distribution of the page load time (seconds) per webpage, averaged over different locations.

high-end vs. low-end phones over the years. To this end, we considered the 100 webpages most frequently visited in 2019. For each page, we retrieved a version per year over the period 2015-2020 from the Internet Archive Wayback Machine [123]. The pages whose versions had technical issues were filtered out, ending up with a total of 55 webpages. We cloned the retrieved versions on our own web server and ran all experiments locally on that same server. This ensures that, when comparing webpages across different phones and years, we eliminate any differences related to network connectivity, access, and servers. For each year in 2015-2020, two mobile phones released in that year were used—a low-end phone and a high-end phone—to access the webpages retrieved in that year. We set up WebPageTest [253] to record the JavaScript processing time while accessing the pages from the different phones.

Fig. 3.2a shows the average time taken over the 55 webpages per year, using high-end phones (blue curve) and low-end phones (red curve); the phone models are named in the figure itself, and their technical specifications are provided in Supplementary Tables 3 and 4. As can be seen, the time spent processing JavaScript has decreased slightly on high-end phones, yet increased significantly on low-end phones over the years (from just over 2 seconds to nearly 8 seconds). Note that the increase is not due to a reduction in the processing power of the low-end phones used in our experiment; see Supplementary Table 3. This suggests that the observed increase is attributed to the webpages becoming more computationally intensive over the years. It also suggests that popular webpages are designed with high processing power in mind, neglecting the less fortunate users who can only afford low-end phones, thereby exacerbating the digital inequality. Finally, Fig. 3.2b shows the percentage of page load time spent on JavaScript processing. As can be seen, in the past three years, the percentage was 20% for high-end phones and nearly 50% for low-end phones.

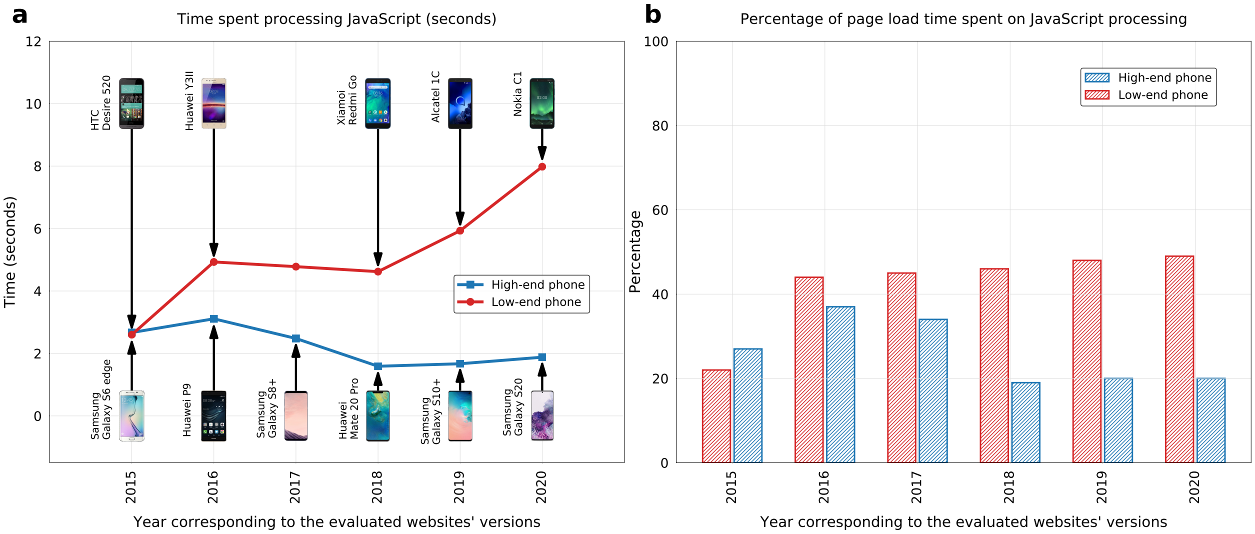


Figure 3.2: JavaScript processing time, measured on high-end vs. low-end mobile devices over the past six years. For each of the 100 webpages most frequently visited in 2019, we retrieved a version per year from 2015 to 2020. The pages whose versions demonstrated technical issues were filtered out, ending up with a total of 55 webpages. For every year in 2015-2020, two mobile phones released in that year were used—a high-end phone and a low-end phone—to access the webpages retrieved in that year; the phone models are specified in the figure. **a**, Average JavaScript processing time (in seconds), measured using a high-end phone (blue curve) and a low-end phone (red curve). The data point for the low-end phone of 2017 was interpolated, since no such phone was available to purchase at the time of the study. **b**, Percentage of page load time spent on JavaScript processing, using a high-end phone (blue bar) and a low-end phone (red bar).

3.2.1 Our solution: Lite-Web

So far, we demonstrated that a significant percentage of page load time is spent on JavaScript processing, and this percentage is greater for users of low-end phones. With this in mind, we propose a solution called Lite-Web, which focuses on producing lighter versions of webpages by optimizing the usage of JavaScript. Lite-Web is a hybrid approach, combining three of our state-of-the-art solutions, namely: *SlimWeb* [57] and *JSCleaner* [55], both of which block non-essential Javascript elements, and *Muzeel* [142], which optimizes essential JavaScript elements. Let us now provide a basic description of these three solutions. For more details on each solution, see Supplementary Notes 2, 3, and 4, and for an overview of related works, see the Discussion section.

SlimWeb is based on the idea that JavaScript elements can be classified based on their code, rather than their serving domains, as is the case with alternative commercial solutions. Relying on JavaScript code is particularly challenging since the code tends to span thousands of lines, and may include obfuscated code (which is deliberately made difficult to understand to prevent reverse engineering), machine generated code (which is often not human readable), or “uglified” code (which is generated via techniques that reduce code size at the expense of readability). By leveraging machine learning techniques, *SlimWeb* is not only capable of overcoming the above challenges, but also classifying previously unseen elements, including unknown libraries, unidentified serving domains, and obfuscated code, all of which are commonly found in today’s Web. Such classification would not be possible using standard profiling techniques. As for the classes used in *SlimWeb*, they are based on the main JavaScript categories identified by experts in the web community [22]. Out of these classes, *SlimWeb* blocks the following three: (1) *Advertising*, which facilitates advertisement; (2) *Analytic*, which collects data about the users; and (3) *Social*, which enables social interactions such as likes and shares.

Having described *SlimWeb*, let us now move on to *JSCleaner*—the second component of our hybrid approach. Specifically, this rule-based solution is used to identify and block non-essential JavaScript elements that do not fall under any of the three classes used by *SlimWeb*. These elements are classified by *JSCleaner* as non-critical to the user experience if their code does not contain any functions that handle the page content or functionality.

Finally, let us describe the third component of our hybrid approach, namely *Muzeel*. Unlike the previous two solutions, which block non-essential JavaScript elements, *Muzeel* optimizes the code of essential elements. This is done by identifying and eliminating *dead code*—parts of the JavaScript code that are never used by the webpage. One of the reasons behind the existence of such code is the use of general-purpose libraries that provide far more functionalities—and hence far more code—than what is actually required by the page. The use of such libraries is a common practice among web developers to speed up the development process, with libraries such as jQuery appearing in 83% of mobile pages worldwide [107]. The identification of dead code is challenging for several technical reasons stemming from the dynamic nature of the JavaScript programming language; see the works by Chugh et al. [61] and Obbink et al. [179] for more details. *Muzeel* utilizes a novel interaction-bot that emulates how a user may interact with the page. Such an approach enables the identification of JavaScript functions that can safely be removed without affecting the user experience and the overall page content.

3.2.2 Evaluating Lite-Web

To evaluate the impact of Lite-Web, we needed to run field experiments that are true to the web browsing experience in developing regions. As a first step, it was crucial to identify a location where the inhabitants' well-being is severely affected by poor Internet connectivity. Moreover, both the websites and the mobile phones used in the experiment needed to be popular in the identified location. Finally, the participants involved in the evaluation were required to be digital natives, who regularly browse the Internet and are familiar with the local network conditions.

Against these desiderata, we chose the Gilgit-Baltistan province in Pakistan, where poor

Internet quality causes severe disruption to students, preventing them from keeping up with their peers. This was demonstrated by the students' protests in July 2020 demanding digital rights [231], leading to the hashtag #Internet4GilgitBaltistan becoming the second-highest ranked on Twitter in Pakistan [2]. As for the mobile phone on which the experiments are conducted, we chose the same low-end phone used earlier, namely QMobile i6i 2020, since it is manufactured by a popular Pakistani company. Finally, we used the Tranco-list [146] to retrieve the 100 Pakistani webpages that were most frequently visited in 2021. Now, we are ready to evaluate the impact of Lite-Web both quantitatively (using automated measurements) and qualitatively (through a user study).

3.2.3 Quantitative evaluation

we sent two teams to four different locations within the Gilgit-Baltistan to measure the impact of Lite-Web based on four evaluation metrics: page load time, Speed Index, page size, and JavaScript processing time. More specifically, the four locations are Taus, Hundur, Sherqilla, and Puniyal, all marked on the map in Supplementary Fig. 4. The measurements were conducted using the WebPageTest framework [253], where the QMobile i6i mobile phone was controlled through a laptop to automatically launch both the original and the Lite-Web versions of each of the 100 Pakistani webpages. This experiment was repeated three times to account for any subtle variations that may arise when the same webpage is visited multiple times. As a result, we ended up with a total of 1,200 visits (4 locations × 100 webpages × 3 visits).

Fig. 3.3a depicts the impact of Lite-Web on page load time—a measure representing the elapsed time from initiation (when the user types in the Web address) to completion (when the page is fully loaded). As shown in the figure, the reduction in page load time across the four locations is 68% (in Taus), 43% (Hundur), 72% (Sherqilla), and 64% (Puniyal), with the average time reduced from 61 to 23 seconds. To determine whether this improvement is sufficient to bridge the digital divide, we compared Lite-Web's outcome to what the people of Gilgit-Baltistan would experience if they were browsing the same 100 Pakistani pages in a developed region (Dubai) on a high-end phone (Samsung Galaxy S20+) using a superior cellular network connection (4G+). As can be seen in Fig. 3.3a, the additional waiting time that users in Gilgit-Baltistan would suffer compared to their privileged counterparts is reduced from 48 seconds (average difference between yellow bars and pink bar) to just 10 seconds (average difference between blue bars and pink bar), amounting to an overall reduction of about 80%.

Fig. 3.3b corresponds to the second performance metric, namely Speed Index, which measures the time taken for the contents of a page to be visibly populated and displayed to the user. Again, the use of Lite-Web results in a significant improvement across all four locations, reducing the gap between developed and developing regions by about 70%. Fig. 3.3c depicts the impact of Lite-Web on the time spent processing JavaScript. As can be seen, the time drops by an average of 54% across locations, and the gap between Gilgit-Baltistan and Dubai drops by about 80%. Fig. 3.3d shows how the size of different webpages is reduced by Lite-Web. Specifically, the page size averaged across webpages and locations is reduced by about 50% (from 0.54 to 0.28 megabytes). Notice that the average page size in Gilgit-Baltistan (without Lite-Web's improvements) is slightly smaller than Dubai's. This is because high-end phones request bigger size images compared to low-end alternatives. However, after using Lite-Web, the webpages become smaller than those downloaded in Dubai by about 60%.

Finally, we evaluated the impact of each of Lite-Web's constituent parts, namely SlimWeb, Muzeel, and JSCleaner. As shown in Supplementary Fig. 5, SlimWeb is the most impactful in terms of the time-based metrics (page load time, Speed Index, and JavaScript processing time), while SlimWeb and Muzeel have a comparable impact in terms of page size reduction.

We compared Lite-Web to two state-of-the-art industry solutions that are widely deployed, namely Opera Mini [182] and Brave [42]. In particular, Opera Mini sends users' webpage requests to their proxy server, where the pages are first requested and then compressed before being sent back to the user in order to reduce the transfer size and speed up the browsing experience. It is

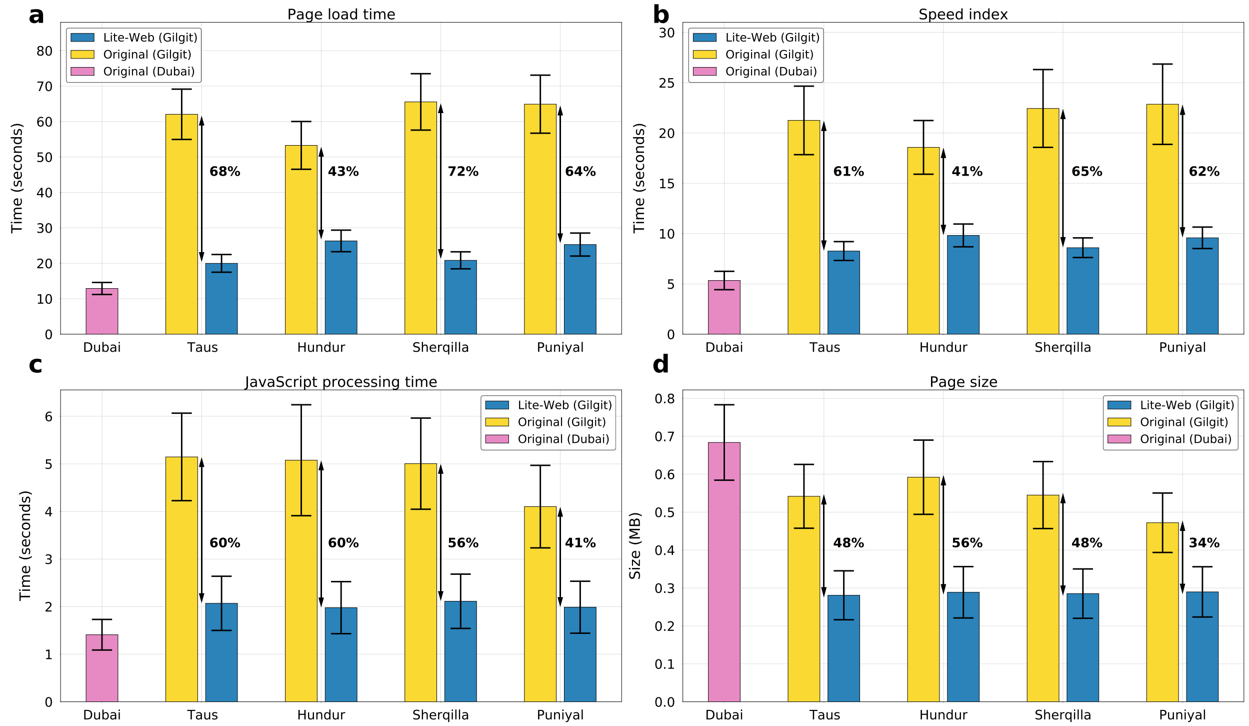


Figure 3.3: Quantitative evaluation of Lite-Web. Using the 100 most frequently visited Pakistani webpages in 2021 to evaluate Lite-Web in four locations situated in the Gilgit-Baltistan province—namely Taus, Hundur, Sherqilla, and Puniyal. The evaluation is done by comparing the Lite-Web version (blue bar) to the original version (yellow bar) on the same low-end phone (QMobile i6i 2020) under the same cellular network conditions (SCOM 4G). Additionally, both the original and the Lite-Web versions are compared to a baseline (pink bar) whereby the same 100 webpages are running on a high-end phone (Samsung Galaxy S20+ 2020) under a cellular network in Dubai (Etisalat 4G+). Error bars represent the 95% confidence intervals. **a**, Evaluating page load time. **b**, Evaluating Speed Index. **c**, Evaluating JavaScript processing time. **d**, Evaluating page size.

estimated that Opera Mini has about 170 million users [**operamini_2**]. However, Opera Mini is prone to breaking interactive sites that rely heavily on JavaScript. Brave, on the other hand, is a privacy-focused browser, which automatically blocks online advertisements and website trackers in its default settings. As of December 2021, Brave has more than 50 million monthly active users, and 15.5 million daily active users [41].

Similar to the evaluation done earlier, we wanted to compare Lite-Web to these two state-of-the-art industry solutions based on four evaluation metrics: page load time, Speed Index, page size, and JavaScript processing time. The measurements were conducted in the city of Lahore in Pakistan using the WebPageTest framework [253], which controlled the QMobile i6i mobile phone to automatically launch the Lite-Web, Opera Mini, and Brave versions for each of the 100 most popular Pakistani webpages. This experiment was repeated three times to account for any subtle variations that may arise when the same webpage is visited multiple times. Note that the results for Opera Mini are only depicted for the page load time and the Speed Index, since the webpagetest framework was unable to collect the remaining two evaluation metrics. The results this evaluation are depicted in Supplementary Fig. 6. As can be seen, Lite-Web achieves improvements ranging between 24% to 57% depending on the benchmark and the evaluation metric.

3.2.4 Qualitative evaluation

To assess whether the above improvements come at the expense of the page look or functionality, we recruited 200 students from two high schools in the Gilgit-Baltistan province. Those students were randomly assigned to control and treatment groups of equal sizes. After that, the 100 Pakistani webpages were assigned to the students as follows: the webpages were divided into 25 disjoint, exhaustive, and equally-sized lists. Then, each list was assigned to 4 randomly chosen students from the control group (who interacted with the original versions of the webpages), as well as 4 randomly chosen students from the treatment group (who interacted with the Lite-web versions). All participants interacted with their assigned versions for 15 minutes using the same low-end phone model (QMobile i6i) equipped with a cellular data connection. Importantly, none of the participants knew the purpose of the study nor the group to which they belonged. This was done to minimize the risk of subject bias, whereby participants tend to behave according to what they believe the experimenter wants to see. The study was conducted by a CITI-trained [63] person following Institutional Review Board (IRB) approval (HRPP-2021-32) from New York University Abu Dhabi. Furthermore, a letter of approval was obtained from the school principal to conduct the study on the school premises. Social distancing measures were observed, and participants were asked to wear masks throughout the study; see Supplementary Fig. 7. As a token of our appreciation, we donated twelve QMobile QTab v7 Pro tablets to the schools' libraries to be used for educational purposes.

The results of the user study are summarized in Fig. 3.4. Specifically, the left panel of Fig. 3.4a summarizes the users' evaluation of the webpages' appearance. This shows no significant difference between the control and treatment groups, In other words, we found no evidence indicating that the performance gains attributed to Lite-Web come at the expense of appearance. Similar results were observed when accounting for the gender and age of participants. The right panel of Fig. 3.4a focuses on the users (in both the control and treatment) who noticed something missing in terms of appearance; those users were asked to assess the impact of the missing components on the browsing experience. As can be seen, the treatment looks very similar to the control, with the only difference being two additional participants (out of 100) who indicated a slight impact of the missing components, and four additional participants who indicated no impact.

Fig. 3.4b is similar to Fig. 3.4a except that it evaluates the impact of Lite-Web on the webpages' functionality rather than appearance. Again, the left panel shows no significant difference between the control and treatment. In other words, we found no evidence that Lite-Web's performance gains come at cost to functionality. Accounting for users' gender and age reveals similar trends. The right panel of Fig. 3.4b focuses on the few participants who noticed something missing in terms of functionality. Five additional users (out of 100) in the treatment group indicated a slight to moderate impact, and three additional users in the control group indicated a high impact. Finally, after participating in the study, all 200 students were asked to indicate the degree to which they agree with the following statement: *I occasionally avoid visiting certain websites because my Internet is too slow to load them.*

Fig. 3.4c depicts the distribution of the responses, showing that the majority (70%) agree (somewhat or strongly) with the statement. These findings suggest that students in the Gilgit-Baltistan province are excluded from certain webpages because of being on the less fortunate side of the digital divide. More broadly, these results suggest that people in developing regions are in need of solutions such as Lite-Web to empower them to reach otherwise practically unreachable parts of the World Wide Web. As a sensitivity analysis, we repeated the same experiment but with a few modifications. First, we divided the 100 websites based on deciles, and randomly picked a website from each part, resulting in just 10 websites. Second, we recruited students from Lahore University of Management Sciences. Third, we recruited 800 participants and asked each of them to evaluate all 10 webpages, resulting in 800 evaluations per webpage. The evaluation yielded broadly similar results.

3.3 Discussion

Our goal was to understand the extent of the digital divide phenomenon worldwide, and propose a scalable and affordable solution that can potentially alleviate it. We measured the mobile data cost and page load time in 56 cities, and found evidence of digital inequality across the globe. In particular, we found the cost of one Gigabyte in some locations to be orders of magnitude greater than in others, and the average page load time to be four times as long. Crucially, in each location, the results were averaged over the same 100 webpages, and the measurements were taken using the same low-end phone model, to unify the experimental setup across locations. An interesting avenue for future work would be to scale up this experiment, covering more areas within countries and over time, to chart the digital divide. Another direction for future work is to extend Lite-Web such that it not only removes deadcode and blocks non-critical JavaScript files, but also identifies and removes potentially malicious JavaScript code from existing webpages, thereby enhancing the users' security.

In an attempt to identify a solution that can bridge the digital divide, we focused on JavaScript elements, which are more computationally intensive than any other equally-sized web component. Specifically, we studied how the above 100 webpages have changed from 2015 to 2020, and found that the time spent processing JavaScript has remained largely the same on high-end phones, but has increased significantly on low-end phones over the years. This suggests that webpages are designed with high processing power in mind while neglecting the less fortunate users who can only afford low-end phones, thereby exacerbating the digital inequality. More importantly, we found that a significant percentage of page load time is spent on JavaScript processing, and this percentage is greater for users of low-end phones.

Motivated by this key observation, we proposed a solution called Lite-Web, consisting of three novel algorithms designed specifically to optimize the usage of JavaScript elements in today's web. We evaluated Lite-Web across four locations in a province of Pakistan known for its poor Internet connectivity, namely Gilgit-Baltistan. The evaluation focused on the 100 most popular Pakistani pages, and was done using a locally manufactured low-end phone. This demonstrated Lite-Web's ability to substantially reduce the size and loading time of webpages, thereby effectively transforming the local browsing experience to that of Dubai's residents who can afford flagship phones with fast Internet connections.

Based on user studies conducted at two high schools and a University in the region, we found no evidence that the performance gains obtained by Lite-Web come at the expense of the look and functionality of the webpages. However, given that Lite-Web blocks ads, it can reduce the revenue of the content providers, and may disadvantage companies in developing regions as they can no longer advertise their services to the users. Having said that, it should be noted that Lite-Web is not the only solution that blocks ads and analytics. In fact, one of the main features of the "Brave Browser" [42]—a very successful modern browser, with more than 50 million monthly active users and 15.5 million daily active users—is to block ads and trackers. Moreover, ads constitute only one of the categories blocked by SlimWeb, which in turn constitutes only one of three components of Lite-Web. If need be, the ad-blocking feature of SlimWeb can be disabled, in which case the solution would still provide significant speedups to the page load time [57].

3.3.1 Limitations

Our study comes with a number of limitations. First, when reporting the page load time and mobile data cost across cities (Figure 3.1), our data represents a single point-in-time snapshot of performance and price. Mobile network performance evolves rapidly, both due to network upgrades as well as increased usage of infrastructure, but these factors are not considered in our analysis. Similarly, we do not consider the role of policy and competitive factors that drive the data cost. Moreover, although participants were instructed to purchase a plan they considered to be affordable, this plan is not representative of the entire spectrum of plans available in their respective city. Having said that, our experiment facilitates a comparison across cities since the price was deemed affordable by an undergraduate student who came from that city (in addition to the experimental protocol which controlled for processing power, web browser, pages visited, connection medium, and time of day). As such, the analysis in Figure 3.1 provides evidence of digital inequality across cities, but should not be interpreted beyond that.

3.3.2 Related work

Over the past decade, expanding Internet access has become a target for international advocacy efforts from the United Nations, and many solutions have been proposed to provide affordable, high-quality connection to everyone. However, such efforts rely on critical infrastructure that would require years to build and hundreds of billions of US dollars to fund [10]. A significantly cheaper alternative is to make the webpages lighter for developing regions. Surprisingly, this alternative has only just started gaining attention. For example, Facebook has introduced a solution called Facebook Lite [85] for Android users with limited connectivity and low-end phones. However, this solution is designed solely for Facebook. Another initiative is Google's Accelerated Mobile Pages (AMP) [93], which provides a framework that can assist web developers in creating lighter versions of their webpages. Unfortunately, AMP does not consider existing webpages, but rather requires the creation of new ones from scratch. This makes it hard to deploy on a massive scale, especially given the billions of webpages already present in the WWW.

From the developer's perspective, one way to reduce the size of JavaScript files before they are embedded into the page is to use uglifiers [35, 62]. These rely on removing non-essential characters such as white spaces and newlines from JavaScript files to improve transmission efficiency. However, unlike our Lite-Web solution, uglifiers do not reduce JavaScript processing time—a major contributor to the digital inequality, as our experiments have shown. From the user's perspective, several JavaScript blocking tools [9, 34, 72, 121, 208] can be used to reduce the amount of JavaScript transferred to their browsers. However, these tools are restricted to a predetermined block-list, and are not equipped with any sort of intelligence that can automatically classify previously-unseen JavaScript elements to determine whether they should be blocked. A very recent solution called Percival [8] has shown promising results in blocking ads using deep learning. It intercepts images obtained during page execution to flag potential ads. However, this solution is computationally intensive, resulting in a non-negligible performance overhead on desktop PCs. As such, it cannot be applied on low-end mobile phones with limited computational power.

In a recent work [172], the authors proposed WebMedic—a method to remove less-useful (rather than entirely unused) functions from the page. They found that 20% of the memory can be saved for the majority of webpages while preserving 80% of the functionality. However, further research is needed to maximize the speedup while minimizing the impact on the page functionality. Other ways to improve the browsing experience are offered by platform-based solutions. For instance, Apple News [21] is a news aggregator app developed exclusively for Apple mobile devices, whereas Instant Articles [85] is a tool that allows publishers to create fast and interactive content on Facebook. However, such solutions are narrow in scope, and are not generalized to all devices and/or all webpages.

3.3.3 Conclusions

We saw how the gap between high-end and low-end phones has increased over the past five years in terms of JavaScript processing. If this trend continues without any interventions, it would lead to a segregation of disadvantaged and advantaged users, whereby the former are practically unable to access the webpages that cater to the latter. Such segregation would violate the *Net neutrality* principle [260], which requires treating all Internet traffic equally, without discriminating or charging differently based on user, content, website, location, type of equipment, or access medium. Our findings call for attention from researchers and policymakers alike, to mitigate disparity and adhere to the net neutrality principle across the globe. More broadly, Internet connectivity has arguably become a basic human right in the twenty-first century, and the emerging literature on reducing web complexity [42, 85, 90, 93, 134, 172, 174, 176, 179, 183, 249, 250] constitutes a promising step towards realizing the United Nation's vision "to ensure that digital technologies are built on a foundation of respect for human rights and provide a meaningful opportunity for all people and nations" [236].

3.3.4 Data Availability

Our data were collected from several experiments that we ran: a) in the wild page load times and cost collected from 56 cities around the world, b) in lab experiments on JavaScript processing times over past six years, and c) in the wild quantitative evaluation of Lite-Web from two schools in Pakistan. The whole data will is available under the following repository https://github.com/comnetsAD/digital-divide.

3.4 Methods

Our proposed Lite-Web solution combines three novel algorithms that we developed to reduce the processing cost of JavaScript in today's webpages, namely SlimWeb [56], JSCleaner [55], and Muzeel [142]. For a given webpage, Lite-Web first runs SlimWeb's machine learning classifier to identify and block JavaScript elements that are non-essential to the user experience; see Supplementary Note 2 for more details. A user study [56] showed that, in order to achieve faster browsing, people are willing to sacrifice parts of the page that are responsible for: (1) advertising, (2) analytics, and (3) social interactions. Based on this finding, all JavaScript elements belonging to the above three categories are blocked by Lite-Web. Additionally, Lite-Web runs a modified version of the rule-based classification used by JSCleaner to identify and block JavaScript elements that are non-critical to the page content or interactive functionality. More details on how Lite-Web modifies JSCleaner's rules can be found in Supplementary Note 3.

So far, Lite-Web preserves JavaScript elements that are identified as essential by SlimWeb and the modified JSCleaner rules. By analyzing these preserved elements, we found many of them to be large JavaScript libraries that are incorporated wholly into the page, even though only a few functions of these libraries are utilized [142]. This key observation suggests that optimizing webpages can go beyond eleminating non-essential JavaScript elements, by optimizing the essential ones. This optimization is done through the elimination of functions that are included in the essential elements yet not used by the webpages. This is precisely what Muzeel is designed to do. The elimination of unused functions provides data cost savings (since the files containing such functions are often quite large), as well as performance improvements [250] (since the number of functions that require processing is now reduced). Further details on how Muzeel operates can be found in Supplementary Note 4.

When evaluating Lite-Web in Gilgit-Baltistan, we deployed Lite-Web in a cloud server hosted in Pakistan. This server maintained a database of JavaScript elements extracted from the 100 most popular Pakistani webpages, labeled by SlimWeb and JSCleaner as either essential or non-essential. The server caches a modified version of the essential ones, which is stripped out of any unused functions by Muzeel. The phones' browsers were configured to utilize our LiteWeb server as a web proxy. As such, JavaScript requests are either deemed non-essential by the proxy and subsequently blocked, or deemed essential, in which case the Muzeel'ed versions of these elements are sent back from the server cache. All other web elements' requests, apart from JavaScript, are served live from the Internet.

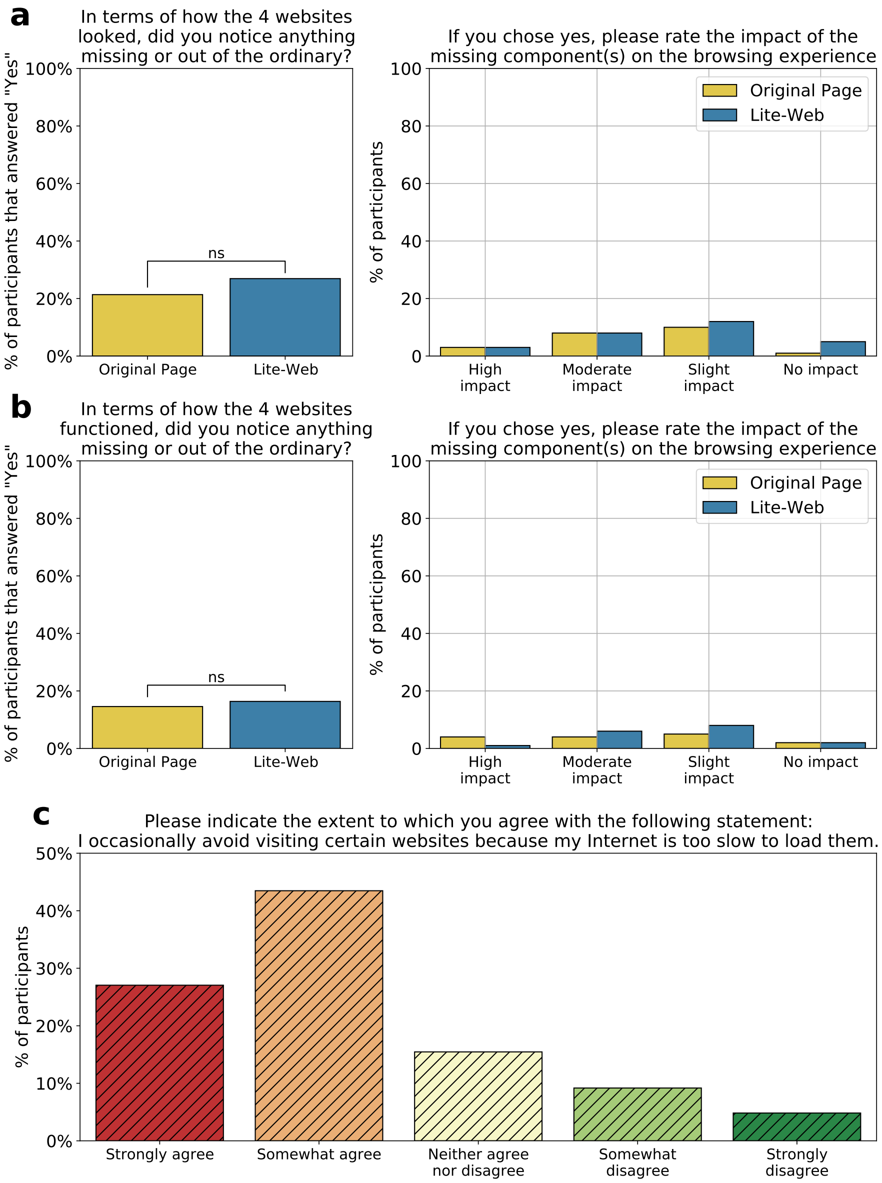


Figure 3.4: High school students' evaluation of Lite-Web's impact on the appearance and functionality of websites. Each participant interacted with 4 of the 100 Pakistani websites most frequently visited in 2021; the control and treatment groups interacted with the original and Lite-Web versions of these websites, respectively. **a,** Left panel: Percentage of participants who answered "Yes" to the question: "*In terms of how the 4 websites looked, did you notice anything missing or out of the ordinary?*" (ns = not significant; p = 0.42); those who answered "Yes" were subsequently asked: "*If you chose yes, please rate the impact of the missing component(s) on the browsing experience*"; the distribution of their responses is depicted in the right panel. **b,** Similar to (**a**) but for questions asking about how the websites functioned, rather than how the websites looked (ns = not significant; p = 0.85). **c,** Responses of all participants (control and treatment) to the question: "*Please indicate the extent to which you agree with the following statement: I occasionally avoid visiting certain website because my Internet is too slow to load them.*"

4 | The Quest for the Best: Evaluating Next Generation Applications in Emerging Mobile Networks

This chapter is adapted from the preprint version of "The Quest for the Best: Evaluating Congestion Control in 5G", submitted to ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT). In this chapter, we perform an evaluation of the prominent state-of-the-art congestion control algorithms currently in deployment to assess their behavior under the unique constraints presented in high-bandwidth and low-latency 5G network environments.

ABSTRACT

The rapid evolution of the next generation of mobile networks has paved the path for the development of a wide array of exciting new applications. Instead of gradual and incremental enhancements, each new generation of mobile networks improves the capabilities of cellular networks tenfold, with 5G typically being 10 times faster than 4G, and 6G is anticipated to improve upon 5G by an even higher factor. Given these rapid exponential improvements, it is important to ensure that all layers of the network stack are able to keep pace and support the performance im-

provements that are possible with these new technologies. Unfortunately, at the transport layer, we still lack a clear understanding of the performance characteristics of current state-of-the-art Congestion Control Algorithms (CCAs) in 5G environments with high channel fluctuations over short timescales. In this chapter, we present *Zeus*, a novel framework that enables unified and repeatable evaluation of CCAs in 5G environments. Secondly, we conduct the most comprehensive cross-protocol benchmarking study to date, covering 10 CCAs across real and synthetic 5G traces, buffer regimes, and application scenarios. Finally, we condense the plethora of raw results that we generate and contextualize them in terms of the performance of CCA in real world applications using a scenario-aware scoring system. Our analysis reveals surprising performance inversions and highlights the important considerations that must guide the design and evaluation of future CCA.

4.1 Introduction

5G networks exhibit several characteristics that make them unique compared to previous generations, such as significantly higher network variability over short time scales [261]. Despite the vast array of CCAs proposed over the past two decades and the large number of studies conducted to measure their performance in 5G [13, 77, 148, 177, 214, 223], we still lack a detailed understanding of the performance characteristics of these CCAs across diverse 5G channels [76, 102, 103, 133, 136, 268, 269]. As a result, a careful study across different dimensions, environments, and buffer sizes is required to shed light on CCAs intricacies, and identify their potential strengths, weaknesses, and trade-offs. Due to the nature of 5G, it is important to verify that the behavior of these CCAs conforms to the expectations set by previous evaluations on non-5G networks. As we find in our evaluation (Section 4.6), this is not always the case. Through this study, we identify the CCAs that are currently best positioned to rise to the 5G challenge and we observe the protocol designs that require modifications to efficiently leverage the performance gains provided by

5G and support the exciting new applications enabled by these improvements.

Type of network	Optimal CCA from throughput	Optimal CCA from delay	Throughput/ delay trade-off	Closest to Oracle	Closest to Optimal	Oracle dist. from optimal
City drive (real 5G)	Cubic, Reno	Copa	Vivace (1 BDP)	BBR	Vivace (1 BDP)	88ms
Beachfront walk (real 5G)	Cubic, Reno	Copa, Vivace, Ledbat	BBR	BBR	BBR (1 BDP)	8ms
Rural Macro (NYUSIM)	Cubic, Reno	Ledbat	BBR (1 BDP)	Cubic (10 BDP)	Reno (5 BDP)	137ms
Urban Macro (NYUSIM)	Cubic, Reno	Ledbat	Verus (5 BDP)	Cubic (10 BDP)	Verus (5 BDP)	137ms
Street Canyon (NYUSIM)	Cubic, Reno	Copa	BBR (10 BDP)	Reno (inf BDP)	BBR (10 BDP)	214ms

Table 4.1: 5G channels summary of results. Distance between the performance of two protocols is measured in terms of throughput/delay trade-off performance.

Metric	Reno	Cubic	BBR	Allegro	Verus	Proteus-P	Vivace	Copa	Proteus-S	Ledbat
Util. (%)	99.7 ± 0.3	99.1 ± 0.4	95.9 ± 0.6	87.6 ± 5.7	84.6 ± 4.7	80.7 ± 3.4	72.9 ± 6.9	63.5 ± 8.2	54.2 ± 7.8	4.9 ± 0.9
Delay	15.6x ± 2.8	12.2x ± 1.5	4.3x ± 0.6	11.4x ± 2.1	67x ± 30	57.4x ± 12.4	3.4x ± 0.5	1.5x ± 0.2	25.8x ± 10.2	1.7x ± 0.3
Intra-fairness	0.88	0.86	0.87	0.84	0.85	0.92	0.87	0.98	0.91	0.91
Inter-fairness	0.81	0.86	0.63	0.64	0.64	0.55	0.66	0.76	0.54	0.58

Table 4.2: CCAs results summary. Utilization is the CCA's channel utilization, delay is the queueing delay, and fairness is the Jain-Fairness-Index (we are measuring for two flows, so the index is lower-bounded by 0.5).

This chapter aims to understand the dynamics of different CCAs in 5G network environments and provide a uniform framework for the evaluation of CCA which can be used to conduct a broad array of CCA evaluations while maintaining the ability to directly compare the results to existing evaluations using the same metrics. The main contributions of this chapter are as follows:

- A comprehensive analysis of the strengths and weaknesses of prominent CCA across various 5G network environments
- A framework for unified real-world and simulation-based experimentation for the analysis of the CCA performance in any network environment coupled with a dataset of network traces collected for these experiments encompassing a diverse range of network environments

For our experimental setup, we perform real-world experiments in the wild using a commercial 5G deployment as the client and a server with different possible configurations of the active CCA. However, real-world experiments present many challenges due to factors including high costs and the high variability in the network environment that renders it difficult to distinguish between changes in protocol performance caused by the environment variability in real time as opposed to protocol performance. We complement these experiments with a wide array of simulation-based experiments that emulate the real-world environments, provide full control over the environment, and are immune to the challenges that hinder repeatability and reproducibility of real world experiments. For the simulation-based experiments, we gather a diverse collection of 5G channel traces, including real-world traces from commercial 5G deployments and trace-driven models built in the NYUSIM model [233]. Finally, we assess 10 prominent CCAs using the enhanced evaluation framework and the collected 5G traces. The chosen CCAs include legacy Internet CCAs like TCP Cubic [100] and TCP Reno, as well as newer state-of-the-art (SotA) algorithms including BBR at Google [46], Ledbat at BitTorrent [209], Copa at Facebook [24], Vivace [80], Allegro [81], Proteus-P [164], Proteus-S [164], and Verus [266]. We compare these CCAs with two reference benchmarks: (i) a *Delayed Oracle Model* where the base station can compute the best congestion response based on the channel state at a given time, but the sender receives this feedback after a propagation delay by which time the channel state may have changed; (ii) an optimal offline throughput and delay computation based on complete knowledge of the channel variability ahead of time. Finally, we introduce a scenario-aware scoring system to compile the results into a simplified metric that informs users how these protocols will perform in real world applications.

Table 4.1 provides a summary of our results showing the best-performing CCAs on different 5G channels, as examined in this chapter. We evaluate CCAs with different buffer sizes, expressed as multiples of the bandwidth-delay product (BDP), based on three distinct metrics: throughput utilization, maintaining low network delays, and achieving a favorable balance between throughput and delay. Table 4.2 presents the average CCA results across the 5G traces, focusing on channel utilization, delay increase relative to the baseRTT (that only accounts for propagation delay), and intra/inter-fairness when competing with another flow over the same link. The CCAs are listed according to their channel utilization, with Reno and Cubic dominating at over 99% utilization. BBR ranks third with approximately 96% average channel utilization, followed by Allegro and Verus. Ledbat exhibits the lowest channel utilization at a mere 5%.

Our results reveal key insights to inform the future of CCA design and reinforce the need of a unified evaluation framework. No single CCA dominates across all conditions. Legacy protocols such as Cubic and Reno continue to outperform many newer designs in throughput-intensive scenarios, while delay-sensitive algorithms like Copa and Vivace excel in real-time and fair-sharing environments. Learning-based CCAs often exhibit instability and poor coexistence behavior in volatile 5G settings. We further find that protocol behavior consistently maps onto a throughput-delay trade-off frontier, where improvements in one dimension come at the cost of the other. This frontier is shaped by buffer size and protocol tuning but cannot be fundamentally circumvented. To make sense of the large number of data points, we introduce a scenario-based QoE scoring system that maps CCA behavior to practical deployment needs, offering protocol recommendations for streaming, real-time, and mixed-traffic environments. This chapter attempts to answer the question: what is the best congestion control algorithm for 5G for different applications and network contexts? Zeus provides a principled and extensible platform to make that answer reproducible, quantitative, and scenario-aware.

4.2 Related work

Evaluating congestion control algorithms (CCAs) has long been a central focus of networking research. New CCAs are typically coupled with an analysis and comparison against a small number of existing protocols under selected conditions. For instance, Copa [24], BBR [46], PCC-Vivace [80], and Proteus [164] each present comparative evaluations, but these tend to be limited to low-bandwidth environments, limited buffer configurations, and a small set of network environments. Controlled network environments often do not reflect the burstiness, queuing dynamics, or RTT variability found in commercial 5G networks. As a result, most prior studies, while sufficient to demonstrate their performance for certain scenarios, these evaluations are often narrow in scope, use inconsistent methodologies, and yield results that are difficult to compare or generalize across protocols and deployment scenarios.

A wide range of simulation and emulation tools have emerged to support congestion control research over the years. Simulators such as NS-2/NS-3 [125, 205] and OMNeT++ [242] are widely used due to their flexibility in defining topologies, transport models, and link-layer behavior. However, they lack execution-time realism and are not designed to run real protocol stacks or real applications. Real-time emulators such as Dummynet [207], NetEm [109], Mininet [104], Hercules [191], and Mahimahi [175] provide more realistic testing by operating with live applications and transport stacks. These tools allow traffic to be shaped at runtime, enabling more representative evaluations of protocol performance. However, they are inherently limited by hardware capacity, making it difficult to scale to multi-Gigabit rates without introducing bottlenecks in replay fidelity or timing accuracy. In particular, Mahimahi supports simple delay and bandwidth variability, but it the original implementation of Mahimahi suffers from bandwidth limitations discussed in 4.4.2. Pantheon [264] was another framework for CCA evaluation through automation and reproducibility. It included a curated set of CCAs and a test harness for benchmarking across a controlled testbed. However, it was primarily designed for legacy TCP protocols and does not support high-throughput mobile environments, modern transport protocols, or cellular traces with realistic 5G dynamics. Moreover, the project is no longer actively maintained and lacks support for extensibility with newer CCAs or application-specific workloads.

Beyond software tools, several 5G-specific testbeds have been deployed in recent years to provide experimental infrastructure for mobile networking research. Examples include 5TONIC [4], 5GIC [138], FOKUS [87], COSMOS [198], POWDER [43], and AERPAW [161]. While these platforms offer access to real hardware, radio resources, and edge-cloud setups, they are often geographically constrained, require application approval, and are generally tailored to specific experiment types (e.g., wireless PHY testing, MEC deployments). As such, they are not designed for reproducible, trace-driven protocol evaluation at scale, and lack the flexibility and accessibility needed for rapid iteration across a large set of CCAs.

In contrast, Zeus is a unified, extensible framework that addresses these tooling and methodological limitations. It enables reproducible, high-throughput evaluation of CCAs using real-world 5G traces. Zeus extends Mahimahi to support the capabilities critical for capturing the high bandwidth capacities, short-timescale variability, and burstiness of 5G. Unlike prior tools, Zeus standardizes the evaluation process, enabling fair, side-by-side comparisons across a diverse set of protocols. This chapter utilizes these capabilities conduct a comprehensive benchmarking study of 10 CCAs across different traces and buffer configurations, and multiple metrics including fairness, delay inflation, and harm to Cubic highlighting the trade-offs relevant for real-world deployment.

4.3 Research Methodology

Assessing these CCAs in 5G environments remains a complex challenge. Factors contributing to the difficulty of this task include the 5G environment's variability due to uncontrollable elements such as competing network traffic, signal fluctuations, and the large propagation losses caused by the high-frequency nature of mmWave signals. Due to such factors, running the exact same 5G real-world experiment at the same location can yield highly variable results. Cost is another significant factor; a single one-minute experiment can consume up to 7 GB of data at a rate of 1 Gbps. This makes the analysis of real networks extremely difficult because one cannot attribute the observed effects only to CCA change, given these factors. As a result, researchers often resort to simulators or emulators. However, when it comes to 5G, access to high-quality prototype frameworks is limited, restricting the ability to test novel CCAs in realistic 5G settings. Given these challenges, it is vital to evaluate and compare various CCAs across diverse 5G channels in a consistent manner. Repeatability and reproducibility are crucial to drawing accurate and meaningful conclusions, as the variability of 5G makes evaluating CCAs in the real world challenging and expensive.

4.3.1 Metrics for Evaluating CCAs

(i) End-to-end throughput and delay: It is reasonable to question whether existing CCAs are compatible with the characteristics of 5G networks and whether their performance is impacted by the unique challenges presented in 5G environments. Specifically, we focus on evaluating the end-to-end throughput and delay in the context of the diverse 5G networks.

(ii) Fairness and Harm: In today's literature, it has become customary to evaluate the fairness of new CCA with respect to legacy solutions when sharing a bottleneck. To determine this, several papers use Jain's fairness index. However, in our analysis, in addition to Jain's fairness index, we also evaluate a more practical harm-based approach, defined in [251]. This approach analyzes the harm that a new CCA causes to an existing legacy CCA when sharing a bottleneck to ensure that new CCA does not cause more harm than existing solutions. We chose Cubic as the legacy CCA, since it is the default TCP flavor on many of today's platforms, such as Linux [256] and Android OS [6]. We measure the harm caused to Cubic flows by other competing CCA flows to determine if the new CCA is suitable for coexistence with Cubic in the wild.

(iii) Impact of Buffer Sizes: A critical aspect of evaluating a CCA in cellular networks is examining the impact of the bottleneck buffer size. Many previous studies have either overlooked this vital parameter or concentrated on relatively small buffer sizes. Therefore, we assess four buffer size settings based on the bottleneck buffer size: 1 BDP, 5 BDPs, 10 BDPs, and an infinite buffer.

4.3.2 Reference Benchmarks

To better understand the challenges 5G environments pose for CCAs, we introduce two reference models to compare against: the *Optimal reference model*, and the *Delayed feedback Oracle model*. The optimal reference model represents the ideal operating point with a fully saturated channel and baseRTT assuming complete knowledge of the channel variability ahead of time. The Delayed Feedback Oracle model represents a hypothetical algorithm situated at the cellular base station with perfect knowledge of the current 5G channel state only at a given time without any future knowledge about the state of the channel. If the sender could access this information and the primary bottleneck was the cellular link, it could fully utilize the link without incurring additional delays. However, since this information is only available at the base station, it must be shared with the sender, introducing one-way delay, followed by another one-way delay for the data to reach the receiver. In theory, this is the best information a CCA can use, but most lack access to it and instead infer it through signals like delays, inter-arrival times, etc.

4.4 The *Zeus* Framework

This section describes *Zeus*, a framework designed and developed to conduct performance analysis of CC algorithms over 5G channels. The analysis methodology described in this section is general and technology-agnostic so that it can be extended to study the performance of CC algorithms over other types of channels. The overall workflow is detailed in Figure 4.1. As for the channel traces, different sources can be used as long as they follow the appropriate format detailed in Section 4.4.2. Furthermore, the framework includes a set of traces obtained from two different sources: a real 5G cellular network of a commercial operator (Non-StandAlone (NSA)) and mmWave traces generated with an ns-3 network simulator. *Zeus* embeds a tailored version of ns-3 [205] to generate traces for mmWave scenarios, using the mmWave module [166]

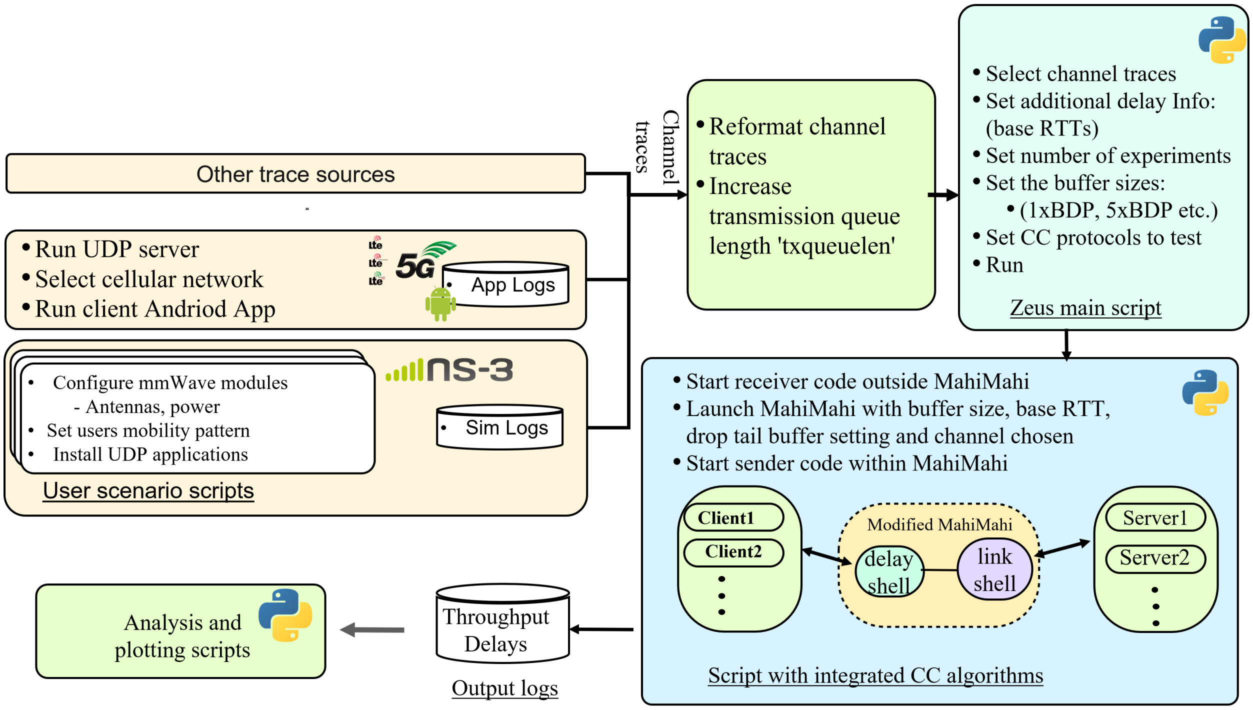
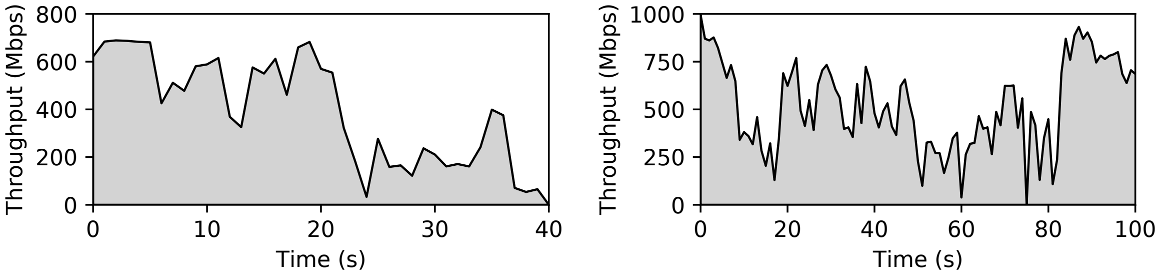

Other trace sources
• Run UDP server
• Select cellular network
• Run client Andriod App
5G
• App Logs
• Configure mmWave modules
- Antennas, power
• Set users mobility pattern
• Install UDP applications
ns-3
• Sim Logs
User scenario scripts
Channel traces
• Reformat channel traces
• Increase transmission queue length 'txqueuelen'
• Select channel traces
• Set additional delay Info: (base RTTs)
• Set number of experiments
• Set the buffer sizes:
• (1xBDP, 5xBDP etc.)
• Set CC protocols to test
• Run
Zeus main script
• Start receiver code outside MahiMahi
• Launch MahiMahi with buffer size, base RTT, drop tail buffer setting and channel chosen
• Start sender code within MahiMahi
Client1
Client2
Modified MahiMahi
delay shell
link shell
Server1
Server2
Script with integrated CC algorithms
Throughput
Delays
Output logs
Analysis and plotting scripts

Figure 4.1: Architecture of the *Zeus* framework

developed by NYU wireless. Figure 4.2 shows 2 of the recorded channels. The link emulator can be tailored with additional end-to-end delay, loss rates, and/or modifying the buffer size of the link. The framework sets up sender/receiver pairs using a CC algorithm and connects them through Mahimahi. The results allow us to compare the behavior of different CCAs in the exact same environment in a systematic manner that is repeatable and reproducible. *Zeus* also allows the analysis of bottleneck link-sharing among traffic flows. It leverages the best out of the real-life and emulation realms without the reproducible limitations.

4.4.1 Real 5G network traces

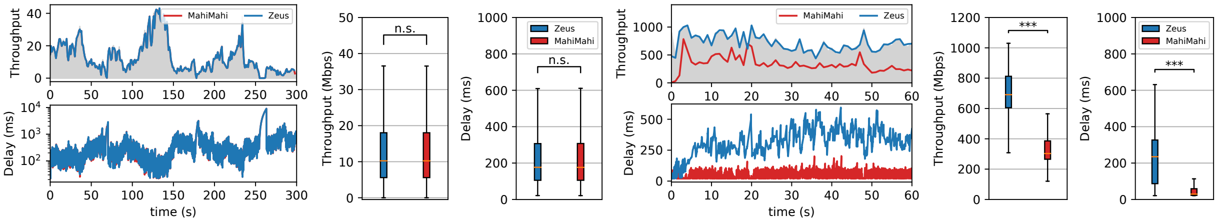
For the generation of real 5G network channel traces, *Zeus* offers a client and server-side implementation that can record real-time 5G channels in a cellular environment. For the trace generation, the server must have sufficient upload capability exceeding the maximum download speeds of the 5G client. The client consists of an Android application that communicates with the



(a) `5G city drive` **(b)** `Street canyon`

Figure 4.2: 5G channel traces

server. Upon connecting, the Linux-based server begins sending UDP packets with a maximum transmission unit (MTU) size of 1500 bytes at a constant data rate to the Android mobile client. The Android application serves as a sink and logs the inter-arrival times of the received UDP packets. These logs are then used to create the trace files, which are converted to the proper channel trace format compatible with the modified Mahimahi emulator.



(a) `Verizon-LTE (4G)` **(b)** `Stat. signif. (4G)` **(c)** `Indoor InHM (5G)` **(d)** `Stat. signif. (5G)`

Figure 4.3: Benchmarking *Zeus* against Mahimahi on 4G and 5G channels

4.4.2 Modified Mahimahi

The original Mahimahi implementation has limitations that affect the emulation of multi-Gigabit capacity channels, such as those based on mmWave. The original implementation has certain bottlenecks that result in a limit of around 400 `Mbps` on the throughput that the network can handle efficiently. Any trace exceeding the threshold of ≈ 400 `Mbps` causes Mahimahi to drop packets randomly, thus capping the channel throughput and utilization. To resolve this, we modify Mahimahi to be able to overcome this limitation. The most critical modification involved drastically reducing the required number of inter-process events for emulating high-bandwidth 5G traces.

Figure 4.3 shows a performance comparison of the modified emulator to the original Mahimahi implementation. The results are obtained by emulating a network flow with consistent full-buffer transmission. Figures 4.3(a) and 4.3(b) show the results obtained when a 4G Verizon-LTE [256] trace was used, with a maximum capacity around `40Mbps`. Figures 4.3(c) and 4.3(d) show the performance obtained over a synthetic 5G channel generated in ns-3 using the Indoor Hotspot Model (`InHM`), with capacity reaching `1Gbps`. Both channel capacities are shown in Figures 4.3(a) and 4.3(c) with the shaded background. It is evident that for the 4G Verizon-LTE channel, both the original Mahimahi and *Zeus* versions offer the same performance, with no statistical significance (n.s.) observed in terms of throughput and delay between the two, completely saturating the channel capacity. However, for the `InHM` 5G channel, the original Mahimahi version fails to handle the number of sending events, leading to the under-utilization of the channel capacity and eventual packet losses due to buffer overflow. In contrast, *Zeus*'s Mahimahi manages to support all the generated traffic, enforcing the correct emulation of the 5G channel, thus mimicking the correct behavior of the CCA in use. A considerable statistical difference for both throughput and delay is observed in Figure 4.3(d) where the achieved throughput for Zeus is significantly higher leading to bufferbloat and higher queuing delays, whereas the exact same experiment with the original Mahimahi implementation is not capable of saturating the available bandwidth.

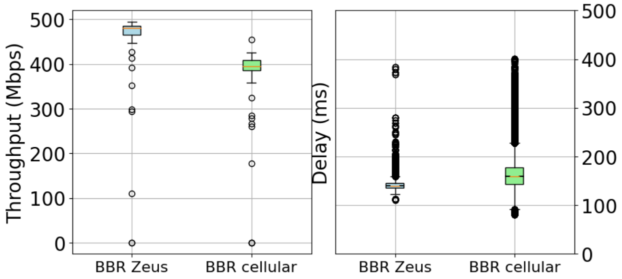


Figure 4.4: *BBR* in a real 5G cellular connection vs. *Zeus*

4.4.3 *Zeus*' operation validation

We validate the correct operation of *Zeus* by comparing its performance with real-world tests. Figure 4.4 shows the results obtained from three real 5G cellular connections, each lasting 60 seconds, using a server running *BBR* and a client device using a real 5G connection in the wild, and results obtained using *Zeus* to simulate a 60 seconds connection using *BBR* over similar network traces recorded at the exact location where the real 5G experiments were conducted. The figure shows that the results obtained with *Zeus* are similar to those obtained from the actual 5G connections, both in terms of average value and sparsity. In particular, the throughput obtained in both cases, *Zeus* and cellular, reach comparable average values and tight distribution. In the case of the delay, the average values are again alike, while the sparsity observed for the *BBR* cellular connection is slightly larger. Although the results are not identical, they serve to ensure that the CC protocols experience similar and comparable conditions with *Zeus* to a real cellular connection. Obtaining identical results is not possible, since the wireless channel realizations are different for the 5G connection running *BBR* and those using UDP to generate the traces used by *Zeus*.

4.5 Experimental Methodology

4.5.1 Real World Experiments

Evaluating the performance of congestion control algorithms (CCAs) in real-world mobile networks poses significant challenges due to the variability and complexity of such environments. While trace-driven simulations offer a controlled means to address many of these challenges, they may not fully capture the intricacies of real-world network behavior. To complement simulation-based evaluations, we conducted a subset of experiments in live mobile network conditions for further testing and validation. The setup involved using a 5G-enabled smartphone as the client device and an AWS EC2 instance in the same geographic region as the server. The server was configured to support five different CCAs via the Linux pluggable congestion control interface.

Using the *iperf3* tool, we performed network experiments in which data was sent from the server to the client using the selected transport protocols. For each protocol, six independent trials were conducted. These experiments allowed us to observe key performance metrics, including throughput and variations in the congestion window size for each protocol. By combining simulation and real-world testing, this approach provides a more comprehensive evaluation of CCA performance, highlighting their behavior under controlled and dynamic network conditions.

4.5.2 Emulation Environment

To create a consistent testing ground for checking the effectiveness of CCAs in a way that is realistic, repeatable, and reproducible, we improved the Mahimahi link emulator [175], which is commonly used to test CCAs in a range of network conditions [7, 24, 80, 94, 264]. This enhancement aims to improve Mahimahi's support for multi-gigabit capacity traces, allowing for the assessment of CCAs without altering their original implementations. The most notable modification involves reducing the number of internal inter-process I/O system calls. In the original implementation of Mahimahi, if multiple packets were to be scheduled simultaneously, Mahimahi would allocate a separate I/O system call for each. In 5G, where capacity can reach multiple Gbps, this can generate hundreds or thousands of system calls at once. We optimize this by introducing a new channel trace format, where multiple packets scheduled at the same time are combined into a single system call. Further details and validation of our framework are provided in Section 4.4.

4.5.3 Generating Realistic Network Traces

We employed two distinct methods for generating the channel traces: i) a real 5G cellular network, and ii) mmWave traces generated with the NS-3 simulator. For the real 5G traces, we used a similar trace collection methodology as [256, 266], with a commercial 5G connection under different mobility scenarios. The details of our approach are mentioned in Section 4.4. Two such traces are used in this chapter, namely: City drive, and Beachfront walking. Finally, the mmWave traces were generated using the mmWave module [166] built atop NS-3 [205], developed by the New York University wireless group (NYU Wireless). We collected two different groups of traces: with buildings and predefined users' motion, and without buildings and with random users' motion. For the first group, we deployed a grid of 3×3 buildings and defined different users' tracks. The Street Canyon comprised two basestations, and users moved following a straight line, getting close or going far to/from each of the base stations. This trace used the urban micro (UMi) 3GPP model. The second group of traces represents open areas. In particular, we exploit the Urban Macro (UMa), and Rural Macro (RMa) 3GPP models to generate the last two traces [271].

4.5.4 CCAs Evaluation Setup

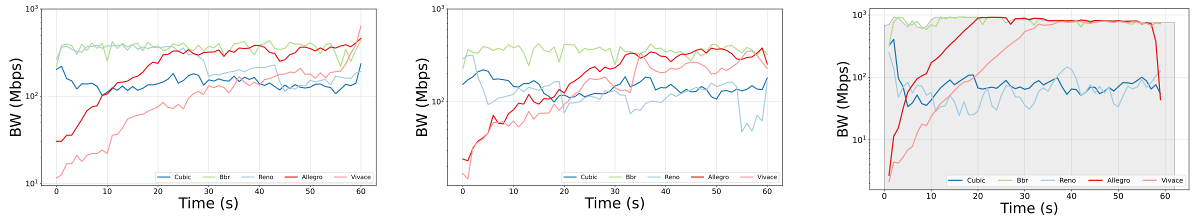
All experiments were conducted on a customized server with an Intel Xeon Bronze 3204 CPU @ 1.90GHz × 12, 15 GiB memory, and Ubuntu 20.04.3 LTS operating system. To ensure accuracy, we consulted with some of the authors of these CCAs to verify their configuration and behavior. For each CCA-channel pair, we performed 5 experiments with four bottleneck buffer sizes namely 1 BDP, 5 BDP, 10 BDP, and infinite buffer.

4.6 Results

4.6.1 Real World Experiments

The throughput trends of the real-world experiments, a subset of which are shown in Figure 4.5, provide valuable insights into the performance of congestion control algorithms (CCAs) under live mobile network conditions. Using a 5G-enabled smartphone as the client and an AWS

EC2 instance as the server, we observed significant variations in key performance metrics across the five CCAs tested. We compare the real world experiment to the results of simulation experiments configured to emulate a similar high-bandwidth real 5G network environment.



(a) Real World Experiment 1 **(b)** Real World Experiment 2 **(c)** Zeus Emulation Experiment

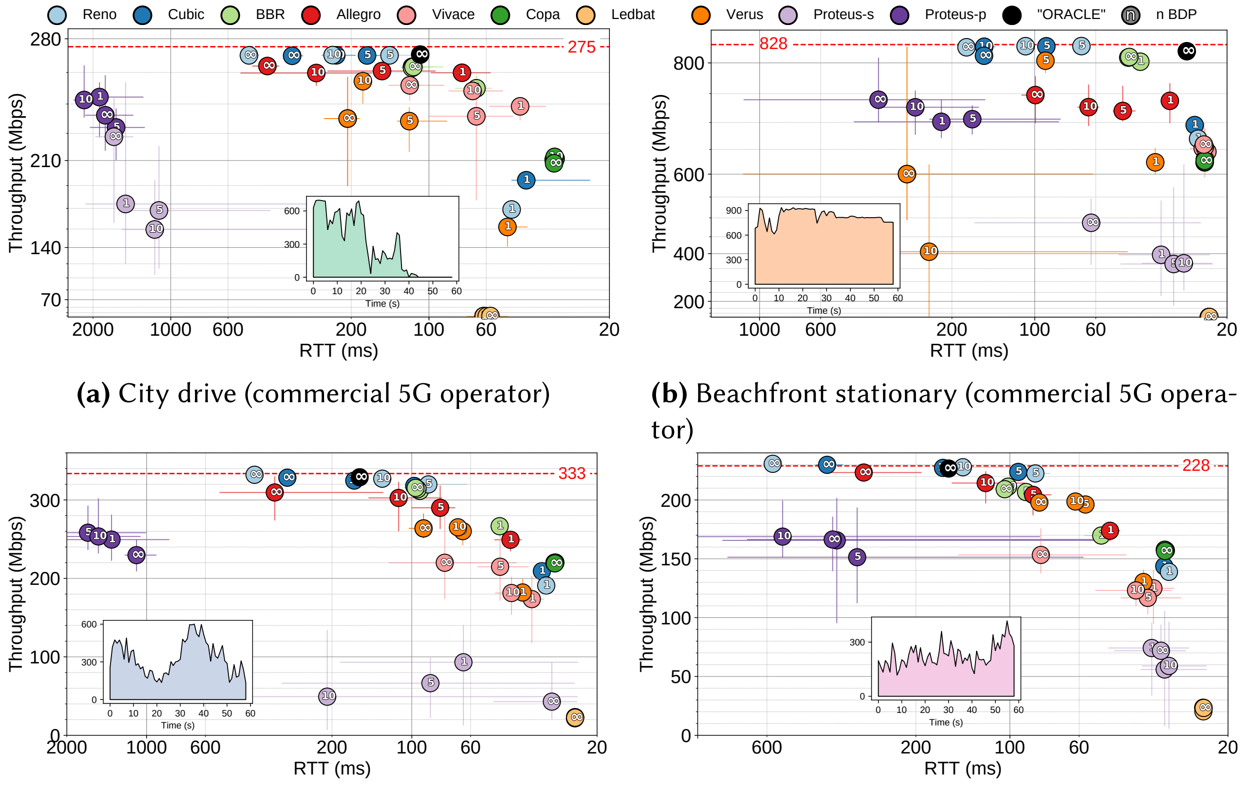
Figure 4.5: Comparison between simulation and real experiments

The throughput results highlight the impact of each CCA's design on its ability to utilize available bandwidth efficiently. We observe that BBR is able to adapt quickly to the volatile high-bandwidth network environment in both cases and consistently achieve high throughput. Legacy CCA are able to maintain high throughput initially but incur bufferbloat which rapidly deteriorates their performance. Allegro and Vivace both gradually converge to their standard operating points in the network environment, which prioritizes high throughput for Allegro, and lower delays for Vivace. Protocols optimized for high-speed environments demonstrated consistently higher throughput, while others exhibited occasional under-utilization due to their more conservative congestion control mechanisms. Packet retransmissions were also observed to vary between protocols, with some CCAs showing resilience in maintaining performance despite sporadic packet drops inherent to mobile networks. Legacy protocols were more sensitive to loss, resulting in a noticeable decline in throughput during challenging network conditions. The replication of the same trends in protocol behavior in the simulation experiments further validate the accuracy of the experimental setup using the simulation framework.

Due to the challenges in real cellular environments, discussed in Section 4.3, it is not sustainable to conduct a comprehensive analysis of CCA at scale relying solely on real world experiments. Thus we introduce the Zeus framework capable of conducting a comprehensive analysis in a repeatable and reproducible manner.

4.6.2 End-to-end throughput/delay analysis

Figure 4.6 illustrates scatter plots that display the performance of ten SotA CCAs over different types of 5G channel traces. The traces include stable channels as well as highly volatile high-bandwidth channels. Each circular data point represents the average throughput (y-axis) and delay (x-axis) performance of a CCA. We conducted 5 separate runs per channel trace for each CCA to ensure the stability of the algorithms' performance and their ability to efficiently use the 5G capacity. Each CCA is indicated by a different color with a number representing one of the four buffer sizes, from as low as 1BDP to an infinite size. The figures also show the capacity over time in an inset subplot.



(a) City drive (commercial 5G operator)

(b) Beachfront stationary (commercial 5G operator)

(c) Rural Macro (NS-3)

(d) Urban Macro (NS-3)

Figure 4.6: CCAs under fluctuating 5G channels

Legacy CCA' performance (Cubic & Reno): Both Cubic and Reno are known for building large sending windows, resulting in a well-known phenomenon called "bufferbloat" in cellular networks which has been well investigated in the context of 3G/4G cellular networks [129]. From our observations across all 5G traces, we found that TCP Cubic and Reno were able to saturate the channel capacity for 5 BDP, 10 BDP, and infinite buffer sizes (the red dotted horizontal line represents the average channel capacity). However, it was observed that a buffer size of 1 BDP was insufficient for loss-based algorithms like TCP Cubic and Reno to saturate the link capacity, as it would lead to packet losses at a faster rate. Our findings also reveal that increasing the buffer size to more than 5 BDP does not necessarily improve channel utilization and mostly results in higher delay. Compared to other CCAs, TCP Cubic and Reno were associated with the highest delays in almost all cases (except for Proteus-P and Proteus-S). Despite legacy TCP being highly criticized given their bufferbloat issues, we show surprising results that they, with reasonable buffer sizes (5-10 BDPs), still outperform many SotA CCAs in some cases, such as the Urban Macro.

Google's BBR performance: BBR's slight under-utilization is due to its operation in a series of states, i.e., the startup phase, a PROBE_BW phase every 8 RTTs to estimate bandwidth, and a PROBE_RTT phase every `10 sec` to re-estimate the minimum RTT. The Probe RTT phase is when BBR briefly reduces its packets in-flight to just four packets, draining any queue build up, which appears to cause the slight under-utilization of the available channel. In our analysis, this design allows BBR to exhibit remarkably consistent performance across all traces, reaching a relatively high channel capacity utilization while slightly underperforming legacy alternatives in terms of utilization. Particularly, its throughput remains slightly lower than the average channel capacity, with 74%-98% utilization for the 1BDP, and 90%-98% utilization for the 5BDPs, 10BDPs, and infinite buffer sizes. Despite BBR's small loss in throughput utilization, it makes up for it in the delay performance, yielding a remarkable reduction in the delays' performance to approximately 50-73% of that of both Cubic and Reno for the larger buffer sizes.

Facebook's Copa performance: Copa also relies on queuing delays for CC and strives to minimize the packet delays by periodically draining the buffers. Copa exhibits stable behavior across all 5G traces, although its efforts to maintain a minimum queuing delay results in the throughput performance being somewhat diminished compared to other alternatives. Copa successfully maintains minimal delays consistently in our analysis. Previous evaluations, conducted under links of up to 100 Mbps [24], show that Copa achieves significantly lower queuing delays, with only a small throughput reduction, outperforming Cubic, BBR, and Allegro. However, we observed that these claims do not necessarily hold in certain 5G environments, where there is a significant throughput penalty associated with Copa. It achieves average utilization between 66% and 76% for all 5G traces. Based on these findings, Copa can be considered an excellent CCA for applications demanding extremely low delays while preserving decent throughput utilization. The results indicate that Copa prioritizes delay over exploiting the capacity.

The PCC Family performance: Allegro operates in a fixed-step increment or decrement when adjusting the sending rate. In 5G environments, these may be too small. This fixed-step sending rate limits Allegro's ability to quickly adapt to changes in the channel resulting in lower channel utilization and increased delays. In our analysis, Allegro attains marginally lower delays than Reno and Cubic, but generally experiences higher delays compared to BBR, particularly in cases with larger buffer sizes like 10BDP and infinite ones. Moreover, Allegro's throughput is generally lower than BBR's but remains higher than Copa's. An intriguing observation from Allegro's results is that increasing the buffer size negatively impacts its performance, especially in terms of end-to-end delay. This is evident in the high-bandwidth stable 5G channels (see Figure 4.6), where expanding the buffer size beyond 1BDP does not result in increased throughput but instead significantly raises the average delay. In fact, for many 5G channels, Allegro performs better with a shallow buffer (1BDP). Another new observation is that Allegro achieves 83% link utilization over a rapidly changing network in previous evaluations [81] but in our 5G traces, this property often does not hold.

Vivace is a learning-based CCA using online greedy optimization methods to control its sending rate via a utility function. However, optimization methods are susceptible to getting trapped in a local optima [155], which is why we see variable throughput performance with repeated experiments. Moreover, Vivace desires to approximate the lowest achievable RTT (RTT_{min}) to decide its sending rate, which explains the steady delays. However, in 5G networks, operating at RTT_{min} is not necessarily optimal. In our observations, Vivace consistently maintains lower delays across many of the 5G traces. Previous evaluations show that Vivace significantly outperforms legacy TCP flavors, Allegro, and BBR in throughput, latency, and friendliness towards TCP [80]. We observe that in our 5G traces, the throughput utilization of Vivace drops significantly. Our fairness analysis shows that Vivace's TCP friendliness in 5G is similar to Allegro and BBR. In terms of throughput, Vivace's performance varies depending on the channel being examined. For instance, in the high-bandwidth stable channels displayed in Figure 4.6, Vivace achieves higher throughput than Copa with almost the same delay. In such situations, Vivace's performance appears to be unaffected by the configured buffer size. However, when assessed over the volatile 5G channels shown in Figure 4.6, it experiences higher delay compared to Copa and benefits from the increased buffer size.

Proteus-S and Proteus-P are built atop an online learning CC framework [81]. Proteus-S, with its scavenger utility, controls the sending rate and leverages delay variation as a sensitive early indication of flow competition. Proteus-S performance varies significantly across channels, either inducing extremely high delays of approximately 600ms and even more for some traces (see Figure 4.6(a)) or exhibiting very low delays for others (see Figures 4.6(d) and 4.6(b)). Likewise, Proteus-S's throughput is also highly variable, generally leaning towards the lower end. Despite having a strategy to compensate for misleading delay variation in a non-congested channel, it exhibits inconsistent performance in 5G due to the uncertainty of the learning-based components responsible for identifying the changes in equilibrium points. On the other hand, Adhering to a scavenger strategy, Proteus-S struggles to adapt to the variability of the analyzed 5G channels.

In contrast, Proteus-P's utilization is higher than Proteus-S's, but it too faces difficulties adapting to highly variable channels, resulting in extremely high delays. Proteus-P controls its sending rate using a modified version of the utility function of Vivace (with negative RTT gradients ignored). These modifications allows it to outperform Vivace in terms of throughput but at the cost of significantly higher delays in 5G. In previous evaluations [164], Proteus achieves 90% utilization when running alone and limits 95th percentile inflation ratio for latency below 10%. We observe that in 5G environments, the performance changes significantly compared to relatively low-bandwidth network environments.

Others: Examining Verus results, it displays varied behavior across all traces, making it unpredictable in 5G environments. The configured buffer size appears to have a significant impact on Verus performance, where increasing the buffer occupancy to larger settings causes the algorithm to create a more substantial bufferbloat. We have excluded the infinite buffer results of Verus in several instances since they were multiple seconds and would have distorted the results visualization. Verus achieves variable performance in 5G. We attribute this to Verus' delay profile curve, which is the basis for adjusting the CWND, struggling to adapt to certain 5G environments. Similarly, contrary to previous non-5G evaluations [209] where Ledbat was able to saturate the link when no other traffic is present, Ledbat is unable to saturate the link capacity in 5G and achieves the lowest throughput across all traces. This is because Ledbat tries to restrict the delays below a predefined target (`default=25ms`), and maintain that value to form a steady-state and never incur delays higher than the target delay. Ledbat's congestion signal is RTT exceeding a target threshold. Due to the variability in 5G environments, this causes Ledbat to achieve low throughput and delay across all channel traces.

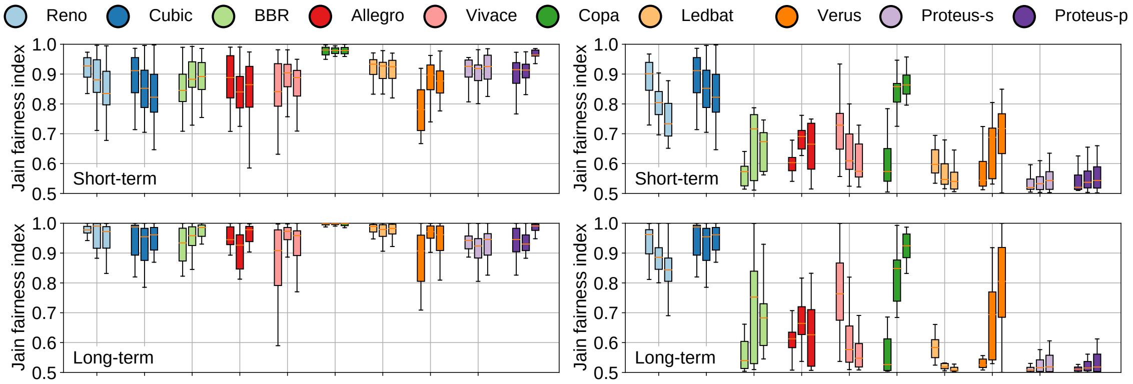
The Delayed Feedback Oracle: The hypothetical Oracle algorithm, having a perfect knowledge of the network conditions, is able to maintain high throughput utilization, saturating the channel capacity across all traces. It is also able to maintain low delays in stable channels. However, we see that in highly fluctuating channels the end-to-end delay suffers a significant negative impact causing it to lose to some of the actual CCAs. This is because when there is a steep drop in the channel capacity, the one-way delay causes the algorithm to be slow to react, thus leading to bufferbloat. This exemplifies the difficulty of creating an optimal CCA for 5G.

4.6.3 Fairness analysis

We evaluate the Jain-fairness-index [126] for all 10 CCAs from two different angles: intra-fairness, where a CCA shares a link with itself, and inter-fairness, where a CCA shares a link with TCP Cubic. Starting with intra-fairness in Figures 4.7(a), the boxplots are divided into short-term fairness (upper subplot) and long-term fairness (lower subplot). These values are computed using a rolling window in steps of $k \times \text{RTT}$, where k is 1, 2, 5, 10, and 20 for short-term and 50, 100, 500, 1000, and 3000 for long-term fairness. For each CCA, three different boxplots are displayed: left for 1BDP, center for 5BDPs, and right for 10BDPs. Nearly all CCAs exhibit decent short-term intra-fairness of 0.8 or higher and even greater long-term fairness of 0.9 or more. In contrast, the inter-fairness results are shown in Figures 4.7(b). Apart from Reno, Cubic, and Copa (for 5 and 10BDPs), all other CCAs demonstrate lower fairness when competing with TCP Cubic, with both Proteus versions and Ledbat having the lowest possible Jain-fairness-index, close to the minimum of 0.5. The primary limitation of the Jain-fairness index is that it measures fairness by assigning equal score in both cases, whether a new CCA utilizes a larger share of the bandwidth than Cubic or vice versa.

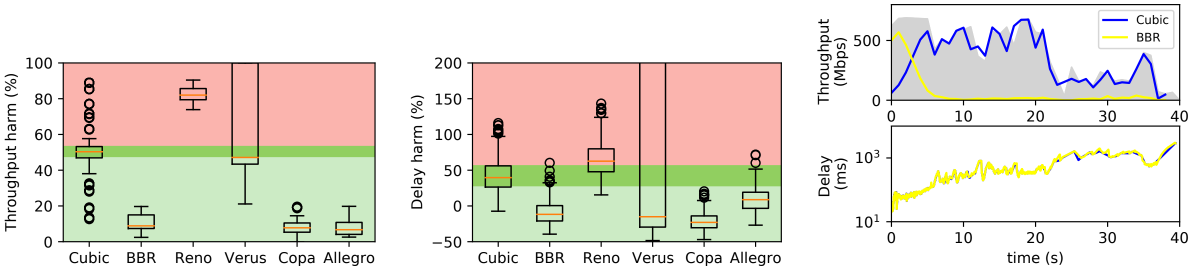
4.6.4 Harm Analysis

A common requirement for any new CCA is its fairness in sharing the bottleneck bandwidth with existing TCP (e.g., Cubic). However, this goal is too idealistic to execute in practice. It is believed that being unfair to Cubic is acceptable because Cubic is not even fair to itself [24]. Inspired by [251], we follow a harm-based approach in this analysis to quantify the harm the



(a) Intra fairness (against itself) **(b)** Inter fairness (against TCP Cubic)

Figure 4.7: CCAs' fairness, each CCA with three boxplots (letf: 1BDP, center: 5BDPs, and right: 10BDPs).



(a) Throughput harm to Cubic **(b)** Delay harm to Cubic **(c)** Instantaneous (Cubic/BBR)

Figure 4.8: Comparison of CCAs' performance over the 5G city drive channel trace with infinite buffer.

CCAs do to Cubic when they co-exist over the 5G channels. We chose BBR, Reno, and Copa for our analysis because of their wide deployment along with Allegro and Verus based on the results of our end-to-end throughput and delays analysis. Following the harm definition in [251], we define x = solo performance; and y = performance after introduction of a competitor connection. Then for metrics where 'more is better' (e.g., throughput) the harm = $\frac{x-y}{x} \cdot 100$. On the other hand, for metrics where 'less is better' (e.g., delay) harm = $\frac{y-x}{x} \cdot 100$.

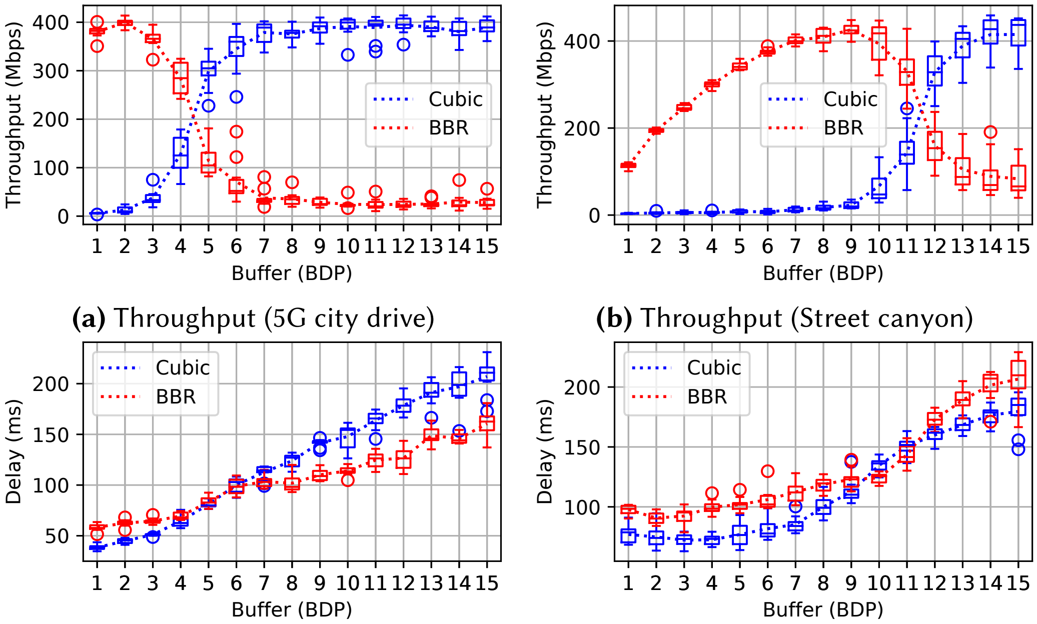
We conducted 20 experiments with an "infinite buffer" for each trace with two Cubic flows co-existing on a 5G channel to calculate Cubic's throughput and delay self-harm as the reference threshold (i.e., the baseline). This threshold provides a firm definition for when a CCA can be deployed alongside Cubic and when it is not. We represent this threshold in the horizontal dark green areas shown in Figures 4.8(a) and 4.8(b). The green area represents the safe zone reflecting the acceptability of the algorithm alongside Cubic when sharing a bottleneck. In contrast, the red zone is where the algorithm is not suitable for deployment. We observe that Cubic causes 50% throughput-harm to itself, reflecting an equal share when competing with itself[1].

In all traces, we observe that BBR, Allegro and Copa fall in the green zone not causing throughput or delay harm to Cubic. However, Reno is the only CC that always falls in the red zone and harms Cubic in terms of both throughput and delay. Reno does more than 80% throughput-harm with 60% delay-harm to Cubic in the 5G city drive trace, above 60% throughput-harm, and nearly 90% delay-harm (similar trends have been observed for other traces). This could be why Netflix adheres to Reno due to its aggressive behavior in dominating other CCAs. However, Google recently announced that Netflix is currently experimenting with BBR [47]. Although BBR is expected to make a major departure from traditional congestion-window-based CC; however, it does not harm Cubic's performance and allows Cubic to take most of the channel capacity across all traces. Figure 4.8(a) shows that BBR does not have a strong impact on Cubic's throughput. With respect to delay, Figure 4.8(b) shows that BBR's impact is slightly higher, but is still below Cubic's self-inflicted harm. We further analyze the interaction between Cubic and BBR in Figure 4.8(c), , where we observe that Cubic dominates the channel capacity, unfairly killing BBR flows in an "infinite buffer" setting.

4.6.5 Impact of buffer sizes

To understand BBR's behavior when sharing a bottleneck link with Cubic and assess the impact of the buffer size on its performance, we experimented with different buffer sizes, varying from 1 to 15BDPs. Figures 4.9(a) and 4.9(c) show the throughput and delay inter-change between BBR and Cubic according to the bottleneck buffer size in the 5G city drive trace. The figures show that at approximately 4.5BDPs both algorithms reach a fair share of the capacity. On the other hand, Cubic claims more capacity upon increasing the buffer size, almost to the point of

[1] Note that negative delay harm indicates a reduction in Cubic delays.



(a) Throughput (5G city drive)

(b) Throughput (Street canyon)

(c) Delay (5G city drive)

(d) Delay (NS-3 street canyon)

Figure 4.9: Effect of buffer size on Cubic and BBR inter-fairness

full dominance around 7BDPs and above. As for values below 4.5BDPs size, BBR dominates the performance leaving almost no share for Cubic to explore.

Similarly, for the street canyon results shown in Figures 4.9(b) and 4.9(d), both BBR and Cubic reach a fair share at around 11.5BDPs before Cubic dominating the channel capacity with growing buffer sizes. We find two logical explanations for this behavior. First, BBR restricts the packets in-flight at a maximum of 2BDPs (the extra BDP deals with delayed/aggregated ACKs). As a result, this extra BDP of data in shallow buffers causes huge packet re-transmissions due to losses. BBR neglects loss as a congestion signal and maintains high re-transmissions over time, in turn worsening things. On the other hand, Cubic adjusts its `CWND` upon a loss. Therefore, BBR causes more packet transmission/re-transmissions than Cubic. This implies that BBR delivers high throughput in shallow buffers but at the expense of high packet re-transmissions. Second, BBR finds the max `target_CWND` as `CWND_gain x BtlBw x RTprop` and increases the window each time an ACK is received until the window reaches the `target_CWND`. Where BtlBw and RTprop are the estimated bandwidth and RTT, respectively. Every 10 secs, BBR probes for `RTprop` by reducing its in-flight packets to just 4 packets to drain the queue. When BBR has an accurate estimate of `BtlBw` and `RTprop`, it caps its packets in-flight at 2BDPs, i.e., BBR allows just 1BDP worth of packets to queue at the buffer for 8 RTTs. Meanwhile, Cubic expands its `CWND` to fill the buffer before encountering loss. Since a flow's throughput is proportional to the buffer share, Cubic gets more packets queued. BBR observes a lower throughput and further decreases its `CWND`. This creates a positive feedback loop allowing Cubic to increase its `CWND` in response to BBR's `CWND` decrease.

4.6.6 Real World Performance

Under the hood, the primary metrics measuring network performance include throughput and latency. However, given the throughput and delay performance of a CCA, it is difficult to extrapolate and compare from these data points to the Quality of Experience (QoE) observed in real applications using these CCA in various network environments. To bridge this gap, we devise a simplified scenario-aware scoring framework that evaluates CCA performance in alignment with practical application demands. Each CCA instance is assessed using a set of normalized metrics: throughput utilization, delay inflation, intra-fairness, and inter-fairness. We define composite utility scores for three representative application scenarios by assigning weights to each metric based on its relative importance. These weights reflect how different classes of applications trade off between responsiveness, utilization, and fairness. They are motivated by a combination of industry standards [3], QoE modeling literature [48], and empirical studies of traffic behavior for these scenarios [37, 143, 156]).

Let x be a CCA instance with normalized metrics Thpt, Delay, IntraFair, InterFair. The following composite score functions are used:

- **Real-Time**: $Score_{RT}(x) = 0.4 \cdot Delay^{-1} + 0.3 \cdot InterFair + 0.2 \cdot Thpt + 0.1 \cdot IntraFair$ Real-time applications such as cloud gaming, AR/VR, and video conferencing are highly sensitive to queuing delay and jitter. To ensure responsiveness, delay inflation and inter-flow fairness are prioritized, with throughput weighted lower to reflect its secondary importance once basic video/audio quality thresholds are met.

- **Large Object Transfer**: $Score_{Bulk}(x) = 0.75 \cdot Thpt + 0.05 \cdot Delay^{-1} + 0.15 \cdot IntraFair + 0.05 \cdot InterFair$ For applications involving cloud backups, and file transfers, sustained throughput is the dominant concern, as buffers and retries can often absorb variable delays. Fairness remains important for consistency across flows, while latency is given moderate weight to discourage extreme delay inflation.

- **Mixed Traffic**: $Score_{Mixed}(x) = 0.20 \cdot Thpt + 0.20 \cdot Delay^{-1} + 0.30 \cdot IntraFair + 0.30 \cdot InterFair$ In enterprise or hotspot environments with diverse traffic types, balanced behavior is critical. Equal weighting reflects the need for protocols that are fair, stable, responsive, and capable of maintaining reasonable throughput under shared conditions.

Scenario	Allegro	BBR	Copa	Cubic	Ledbat	Proteus-P	Proteus-S	Reno	Verus	Vivace
Real-Time	0.510	0.594	0.707	0.588	0.457	0.451	0.377	0.575	0.473	0.555
Large Object Transfer	0.819	0.893	0.695	0.919	0.232	0.772	0.572	0.923	0.795	0.725
Mixed Traffic	0.637	0.688	0.782	0.731	0.574	0.606	0.551	0.719	0.619	0.664

Table 4.3: Scenario-based composite scores for each CCA

This scoring system computes average scores for each CCA based on the analysis results. These composite scores provide a scenario-aware ranking of protocol suitability, allowing us to link observed performance trends to real-world deployment implications. Table 4.3 presents the comparative results across the three application profiles. We observe that Copa is expected to perform well in low-delay and high-fairness environments, aligning with the needs of interactive and real-time applications. In contrast, Cubic and Reno dominate in high-throughput scenarios due to their aggressive sending behavior, though their delay performance is poor. BBR emerges as a strong all-around performer, balancing throughput and delay effectively. Ledbat underperforms in high throughput scenarios due to conservative bandwidth usage. Proteus-S and Vivace also perform well in balanced settings, maintaining fairness and delay control without severely compromising throughput.

4.7 Discussion and Key Takeaways

Our comprehensive evaluation presents an overwhelming set of data points across a diverse set of 5G conditions, CCAs, and buffer regimes. We distill these findings into several core insights that inform both practical deployment and future research directions.

Cubic and Reno, though dated, continue to perform well in terms of raw throughput, especially under moderate-to-large buffer conditions making them effective choices for throughput-intensive applications such as video streaming, cloud backups, and software updates. Notably, Reno sometimes outperforms modern algorithms in some 5G traces, particularly those with moderate BDP and relatively stable variation. Copa and Vivace are strong candidates for realtime, interactive applications such as AR/VR, cloud gaming, and video conferencing. However, while suitable for latency-critical workloads, they may be unsuitable for bandwidth-intensive tasks unless paired with adaptive tuning. Learning-based CCAs like Proteus and Verus are designed to adapt dynamically to network conditions, but their real world performance under 5G variability is inconsistent. Verus displays unstable delay and low throughput in some scenarios, and Proteus variants show erratic responsiveness that leads to both underutilization and high coexistence harm. These findings suggest that while learning-based designs hold promise, current approaches may require more robust mechanisms to handle the abrupt dynamics of mobile 5G networks. BBR and Allegro offer a balanced approach, making them suitable for mixed traffic environments such as enterprise WLANs, shared mobile hotspots, or urban 5G deployments. BBR maintains competitive throughput while moderating delay and limiting coexistence harm, making it a reasonable default for general purpose use. Allegro performs similarly well across most conditions, but like Copa, it occasionally underutilizes links in highly dynamic environments where capacity shifts faster than its reaction time. Both protocols stand out for their ability to navigate multiple performance trade-offs without leaning too heavily toward a single objective.

Across all scenarios, we consistently observe that protocol behavior falls along a clear throughput-delay trade-off curve. Each CCA implicitly chooses a point along this frontier: some favoring utilization at the cost of queuing delay, others sacrificing bandwidth for responsiveness. Tuning protocol parameters or varying buffer size shifts a CCA's position along this curve, suggesting that the set of achievable trade-offs is bounded. This insight reinforces the importance of designing CCAs that can dynamically reposition themselves along the frontier in response to shifting network or application constraints.

In conclusion, our detailed evaluation of ten different CCAs in 5G environments unearthed several limitations of individual protocols without a clear winner across all environments. Even the best performing protocol in specific 5G settings exhibits limitations across other 5G environments including serious fairness concerns when competing with other flows in 5G. Additionally, we present an evaluation framework for CCA to enable various types of CCA evaluations in a unified setting with access to standardized benchmarking that is directly comparable to other CCA evaluations using the standard framework. We believe that there still exists an essential gap in congestion control research in 5G and designing an optimal CCA to address the 5G challenge remains an open research question. Finally, these results highlight the broader value of Zeus as a benchmarking platform. By enabling standardized and reproducible evaluations across diverse 5G scenarios, it could serve as a foundational tool for the design, testing, and selection of future congestion control algorithms that meet the demands of next-generation networks.

5 | Towards Next Generation Immersive Applications in 5G Environments

This chapter is adapted from the preprint version of "Towards Next Generation Immersive Applications in 5G Environments", submitted to ACM Conference on Embedded Networked Sensor Systems (Sensys). In this chapter, we design a QoE-aware modular framework for next-generation immersive applications, comprising a high-level streaming and synchronization layer for AR/VR systems and a QoE-aware rate control protocol optimized for collaborative Extended Reality applications.

ABSTRACT

The Multi-user Immersive Reality (MIR) landscape is evolving rapidly, with applications spanning virtual collaboration, entertainment, and training. However, wireless network limitations create a critical bottleneck, struggling to meet the high-bandwidth and ultra-low latency demands essential for next-generation MIR experiences. This chapter presents Hera, a modular framework for next-generation immersive applications, comprising a high-level streaming and synchronization layer for AR/VR systems and a low-level delay-based QoE-aware rate control protocol optimized for dynamic wireless environments. The Hera framework integrates application-aware streaming logic with a QoE-centric rate control core, enabling adaptive video quality, multi-user

fairness, and low-latency communication across challenging 5G network conditions. We demonstrate that Hera outperforms existing state-of-the-art rate control algorithms by maintaining up to 66% lower latencies with comparable throughput performance, higher visual quality with 50% average bitrate improvements in our analysis, and improved fairness. By bridging the gap between application-level responsiveness and network-level adaptability, Hera lays the foundation for more scalable, robust, and high-fidelity multi-user immersive experiences.

5.1 Introduction

The landscape of Multi-user Immersive Reality (MIR) technology, encompassing both Virtual Reality (VR) and Augmented Reality (AR), is undergoing a transformative period driven by substantial investments from industry leaders like Meta and Apple [115–118, 238]. With the rapid growth of high-bandwidth MIR applications, the demand for ultra-fast and low-latency wireless communication has surged across both indoor and outdoor environments. In indoor settings, high-performance MIR applications often rely on Wi-Fi technologies such as IEEE 802.11ad (WiGig) and IEEE 802.11ay, which operate in the 60 GHz frequency band to enable multi-gigabit data rates. Similarly, outdoor MIR applications increasingly leverage 5G networks, particularly millimeter-wave (mmWave) 5G, to support real-time streaming and interaction. However, despite their potential for high-speed wireless connectivity, both Wi-Fi and 5G networks present fundamental challenges, including limited range, susceptibility to signal attenuation, and high bandwidth variability, all of which threaten the seamless performance of MIR applications.

A critical yet often overlooked aspect of the infrastructure supporting multi-user immersive reality (MIR) applications illustrated in Figure 5.1 is whether existing systems, including transport protocols and application-level streaming architectures, can deliver the Quality of Experience (QoE) required for seamless interaction in dynamic wireless environments [26, 27, 76]. Despite advances in mobile networking, the performance of MIR applications remains constrained

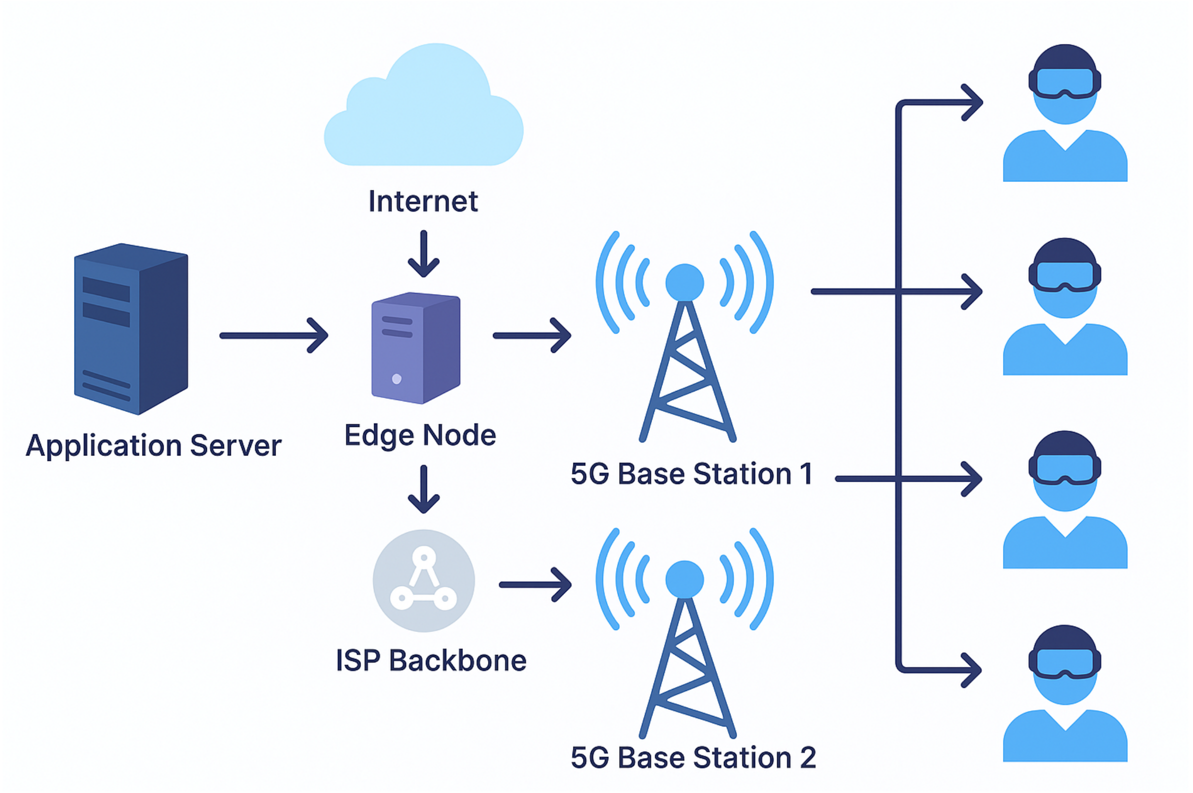


Figure 5.1: Real-world MIR application ecosystem. The end users can be indoors or outdoors and are connected to the application server by a 5G wireless channel.

not only by the limitations of traditional congestion control protocols but also by the lack of integration between network adaptation mechanisms and the application logic responsible for streaming, synchronization, and rendering. Addressing these issues requires a holistic framework that not only optimizes rate control at the transport level but also enables the application to dynamically adapt to changing network conditions to preserve the user experience.

QoE Metric	Description
Startup delay	The time users wait before the immersive content begins. Higher startup delay reduces perceived responsiveness at session start.
Video bitrate / resolution level	The visual clarity of the XR environment. Lower throughput forces the system to lower resolution or increase compression, reducing visual fidelity.
Stall / buffering events	Interruptions in the immersive experience where video or scene rendering pauses to rebuffer. Caused by throughput falling below content rate requirements.
Interaction latency	Delay between a user's action (e.g., moving an object) and the visible response in the shared scene. Directly impacts perceived interactivity and collaboration smoothness.
Collaborative fluency index	The smoothness and synchronicity of shared actions among users, largely driven by latency. High latency leads to disjointed collaborative interactions.

Table 5.1: MIR QoE metrics and their user impact.

5.1.1 Challenges

5.1.1.1 Application Requirements

Currently available MIR applications showcase some of the potential applications enabled by MIR but the Quality of Experience (QoE), in terms of key metrics summarized in Table 5.1 such as video resolution, interaction latency, and collaborative fluency, for these applications is restricted by various network performance bottlenecks [152, 190, 230]. To circumvent these bottlenecks, many of these applications only support 2D experiences, which create a large "screen" in the immersive environment. Others that provide an immersive experience are forced to reduce the frames-per-second (FPS) for the application, which reduces how smooth the application feels. Alternatively, the application displays the content at a lower resolution, which reduces the graphical fidelity, or freezes momentarily, breaking the users' immersion [239]. For a high-fidelity and comfortable MIR experience, a target frame rate (FPS) of at least 120 FPS is recommended, as lower frame rates can cause motion sickness due to the increased discrepancy between visual input and vestibular perception [248]. At the same time, 60 FPS and 90 FPS are often considered as a baseline and target for today's MIR systems [51, 235]. In terms of resolution, a target video quality of at least 4K resolution per eye [229] is required to mitigate the "screen-door effect," a common VR artifact where inter-pixel gaps become visible and compromise the visual fidelity [60].

A high-quality, immersive, multi-user immersive reality experience featuring 360-degree 3D video at 120 frames per second (FPS) and 4K resolution per eye necessitates a downlink bandwidth of at least 100 Mbps per user to accommodate the high data rate associated with 360-degree video capture, stereoscopic 3D rendering, and real-time multi-user synchronization. Furthermore, end-to-end latency must remain below 20 ms to ensure a comfortable and responsive user experience, minimizing motion sickness and maximizing the sense of presence. This latency budget encompasses both network transmission delays and any processing overhead. Conventional congestion control algorithms, like BBR [46] and TCP [15], are often insufficient to satisfy these stringent requirements. While UDP is commonly used for real-time communication due to its low overhead and minimal latency, many multi-user immersive applications require reliable delivery to ensure consistency across users, prevent visual artifacts, and maintain the integrity of complex shared scene states. TCP becomes essential in scenarios where packet loss or out-of-order delivery could disrupt synchronized interactions, collaborative object manipulation, or the seamless rendering of high-fidelity visual elements. BBR, while designed to maximize throughput, can exhibit performance degradation in lossy environments and may not consistently deliver the low latency necessary for VR. Cubic [100], optimized for TCP flows, is similarly challenged by the real-time nature of VR streaming, as its congestion control mechanisms can introduce unaccept-

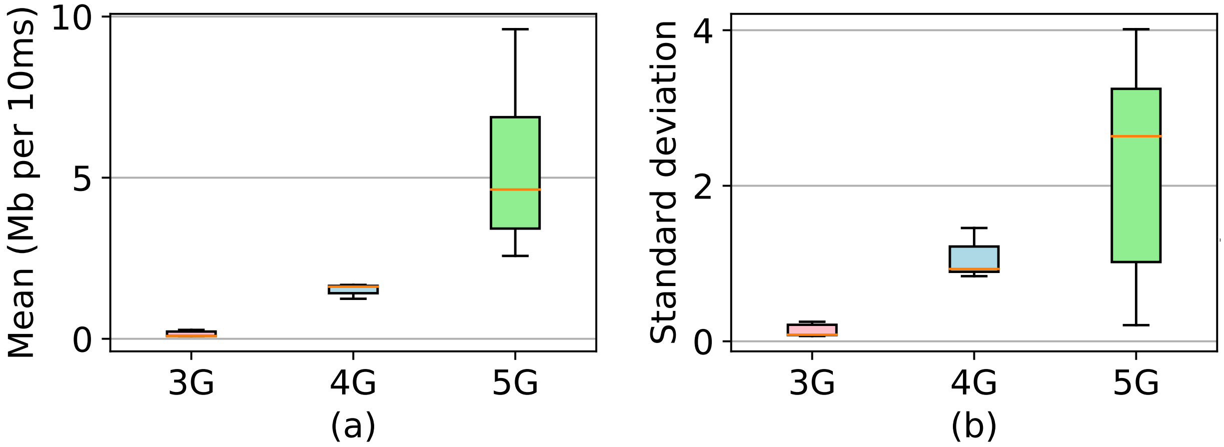


Figure 5.2: Cellular channels' variability analysis. The 5G channel provides higher mean bandwidth, but it comes at the cost of higher bandwidth fluctuations. Data generated through experiments using real-world commercial networks.

able delays, particularly in dynamic network conditions. The high bandwidth demands and strict latency constraints of high-fidelity VR necessitate the exploration and development of specialized network protocols and rate control algorithms tailored for real-time media streaming and interactive applications.

5.1.1.2 5G Channel Variability

Link capacity variation is a known fact in cellular networks. However, the nature of the 5G-New Radio (NR), with a sub-6 GHz spectrum and mmWave bands being vital elements, may cause link capacity fluctuations to be further amplified. Indeed, higher frequency bands suffer from larger propagation losses, effectively reducing the base stations' coverage areas. For instance, due to weak diffraction ability, mmWave communications are sensitive to blockage by obstacles (e.g., humans, furniture, foliage). On the other hand, 5G networks are intended to support much higher mobility scenarios, supporting vehicular speeds up to 500 km/h. On the other hand, line of sight (LoS) and non-line of sight (NLoS) communications experience significantly different channel conditions and throughput. Although 3G and 4G also suffer from these channel fluctuations, these are far more accentuated in 5G links, as illustrated in Figure 5.2 which provides perspective on the sheer difference in both bandwidth between 5G and non-5G networks and perhaps the bigger challenge, the high variability in 5G network environments that make it difficult for congestion control protocols to adapt. Thus, despite the potential to enable high-throughput communications, the variability of the access channel capacity would result in a degraded TCP goodput and very low radio resource utilization. Existing congestion control protocols are commonly designed as general-purpose protocols that consistently perform well in a wide variety of common scenarios. In MIR, however, applications have strict network requirements for optimal performance.

5.1.2 Contributions

In order to overcome these constraints, we propose Hera, a modular framework for MIR applications, built around a novel QoE-aware rate control protocol optimized for next-generation wireless environments. The Hera framework integrates two layers: a high-level streaming and synchronization layer tailored for AR/VR applications and a low-level congestion control core that provides real-time network adaptation through delay-based window modulation and histogram-based RTT tracking. Together, these layers enable the system to dynamically adjust media quality, viewport streaming, and collaborative synchronization rates in response to network conditions, bridging transport-level adaptation and application-level quality of experience. Our work makes several key contributions.

- We design and implement an open-source framework that integrates transport-level rate control with application-level multi-user synchronization to support bandwidth-intensive AR/VR benchmarking. Our implementation connects a custom TCP-based kernel congestion control module with a WebXR-based multi-user application that synchronizes positional updates and interactions across both real headsets and synthetic clients. The framework is designed to expose network metrics to the application layer, providing a foundation

Protocol	Average Quality	Fairness Index
BBR	34.5	0.989
Cubic	12.2	**0.999**
Allegro	45.7	0.370
Vivace	46.5	0.803
Hera	**91.2**	0.965

Table 5.2: Performance comparison of congestion control protocols. Hera achieves the highest quality while maintaining high fairness between competing flows in MIR applications deployed over a 5G network.

for future extensions that could enable adaptive bitrate, resolution, field-of-view streaming, and dynamic synchronization rates.

- Our evaluation goes beyond traditional congestion control metrics by demonstrating how improvements in throughput, latency, and fairness translate into enhanced QoE for immersive applications. We show that Hera maintains low startup delay and stall frequency, sustains higher resolution levels, lowers interaction latency, and improves collaborative fluency compared to existing protocols such as BBR, Allegro, Vivace, and Cubic. These improvements directly impact the comfort, immersion, and usability of AR/VR systems.

- We demonstrate the performance improvements offered by the Hera rate control protocol using a system designed for adaptive video streaming to VR headsets, with a focus on measuring Quality of Experience (QoE). The system supports experiments with both real headsets and synthetic clients, enabling high scalability. It integrates a Linux server hosting DASH content via NGINX, VR and synthetic clients using `dash.js` for playback, and dynamic switching between rate control protocols. This setup allows controlled benchmarking of QoE under various load conditions and network environments. Our experiments show that Hera consistently outperforms other rate control protocols in delivering higher average video quality and smoother playback in multi-user immersive scenarios.

5.2 Related Work

AR/VR Applications in 5G Environments

Many multi-user immersive reality (MIR) applications have been publicly released, but their operation within current network infrastructures often requires compromising Quality of Experience (QoE). For example, virtual social platforms like VRChat [245] and Rec Room [199] enable users to interact in user-generated virtual worlds but face noticeable latency and graphical limitations. Collaborative design tools such as Spatial [225] and Arkio [23] support remote teams co-designing in shared 3D spaces, though they struggle with complex models and large user counts. Immersive training simulations for medical and industrial applications offer hands-on virtual practice but often require simplified visuals and limited interactivity to maintain real-time performance. Large-scale virtual events, such as concerts in Meta Horizon Worlds [165], highlight both the promise of MIR applications and the challenges in delivering high-fidelity experiences to large audiences. Research addressing these challenges includes techniques like viewport prediction [262], adaptive bitrate algorithms [162], FOV streaming [113], and resource-efficient multi-user AR frameworks like Spear [99]. Despite these advances, scaling MIR applications to support seamless multi-user experiences in dynamic 5G environments remains an open problem.

Next-Generation Congestion Control Algorithms

Numerous congestion control protocols have been proposed to overcome TCP's limitations in modern wireless environments. Machine learning-based solutions such as Remy [255], Indigo [264], Aurora [127], and Orca [7] aim to dynamically adjust sending rates under complex conditions, though they often struggle with generalization or stability. Delay-based protocols like Copa [24], Verus [266], and Sprout [257] attempt to balance throughput and latency using end-to-end delay profiles or stochastic models, but typically require tight sender-receiver coordination. Real-time communication protocols such as Google Congestion Control (GCC) [49], ScReAM [131], and NADA [270] are tailored for low-latency media but often exhibit limitations in bandwidth stability or infrastructure dependency. Meanwhile, transport innovations like QUIC [145] and deep learning-driven systems like DeePCCI [216] introduce further flexibility with encrypted header support and pluggable congestion control. Unlike these approaches, Hera integrates histogram-based RTT tracking with probabilistic window adjustment, targeting the specific demands of high-throughput, low-latency multi-user immersive applications in 5G environments.

QoE Metrics and Rate Control for AR/VR

Understanding and optimizing QoE for AR/VR applications has led to the development of frameworks like VR-EXP [86] and Perceive [68], which provide controlled environments and predictive models to assess streaming performance under variable network conditions. These platforms focus on metrics such as startup delay, stall frequency, and frame rate stability. Kulkarni et al. [140] evaluated Wi-Fi configurations for immersive video, highlighting key parameters that affect streaming QoE. While these works address aspects of AR/VR performance, they often stop short of directly linking transport-layer behavior to application-level QoE, particularly in the context of rate control mechanisms that can dynamically adjust to fluctuating bandwidth while maintaining fairness and responsiveness across users, a gap that Hera aims to fill.

5.3 Hera Design

5.3.1 Framework Design

The Hera framework is designed to provide end-to-end optimization and evaluation for multi-user immersive reality (MIR) applications operating over challenging wireless networks such as

5G.

High-Level AR/VR Application Layer

Our high-level application layer is built around a flexible WebXR-based multi-user environment that supports real-time synchronization of user actions, such as positional updates and basic object interactions. This layer is designed to operate with both real VR headsets (e.g., Meta Quest, Pico, or Apple Vision Pro) and synthetic clients that emulate WebXR sessions. Real headsets allow us to validate performance and user experience under practical conditions, while synthetic clients enable large-scale, controlled benchmarking with customizable behavior profiles.

The system currently employs WebSockets for multi-user state synchronization, allowing position updates and interactions to be shared across clients with low latency. The application can be extended to incorporate WebRTC/WebTransport channels for bandwidth-intensive media delivery, adaptive bitrate management, and viewport-optimized streaming in future iterations. A key feature of the framework is its utility as a benchmarking tool. By modifying the sending rate, payload size, and update frequency of synthetic clients, the system can emulate a wide range of bandwidth-intensive applications, including high-fidelity video streaming, collaborative 3D design, and large-scale virtual events.

The Hera framework integrates seamlessly with our custom QoE-aware rate control protocol, implemented as a Linux kernel pluggable module. All application traffic, including multi-user state updates, synthetic client payloads, and configurable benchmarking streams, is transmitted over TCP flows managed by this module. This setup allows the selected rate control protocol to directly control sending rates and adapt to network conditions in real time. The framework is designed to expose network metrics and protocol feedback to higher layers, enabling future integration of congestion control feedback into adaptive application logic. Figure 5.3 provides an overview of the framework components and their interactions.

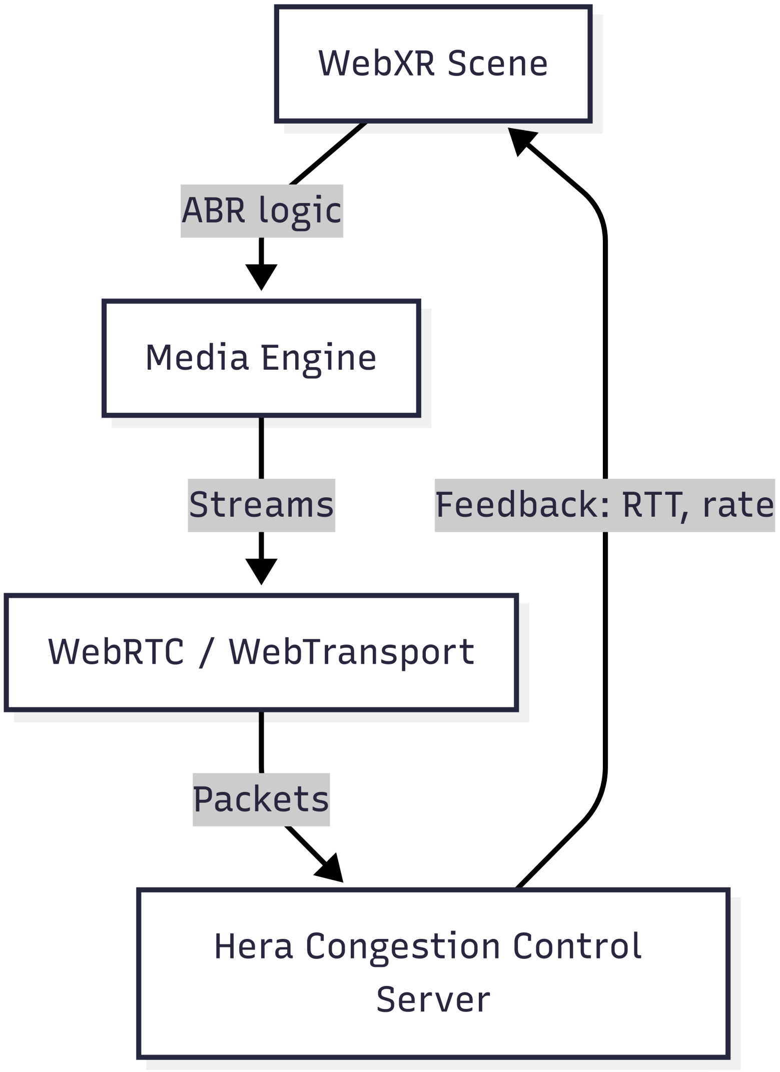


Figure 5.3: Framework architecture showing interactions between components.

5.3.2 QoE Aware Rate Control

5G Network Setup

The Hera rate control algorithm is designed for low-latency applications that operate in a 5G network environment where end hosts communicate with base stations over mmWave channels that exhibit high variability over short time scales [196]. A critical requirement for congestion control in highly volatile network environments is that the protocol must adapt and converge towards the desired sending rate as quickly as possible, given that the environment state can change significantly within only a few RTTs. Given this requirement, the protocol design must consist of a simplistic algorithm that can make the required computations within microseconds while retaining high accuracy of predictions. Any organization deploying a 5G network will deploy several base stations within an area and operate a Radio Access Network (RAN) that can tightly manage all the radio allocations across base stations and end-hosts that connect to the network. We envision a setting where future 5G networks with edge compute infrastructure can support a broad array of low-latency applications with end-to-end latencies of less than $1 - 10$ms. In this regard, we assume that the server endpoint that connects to a low-latency application on a 5G mobile device is within the 5G network or within close network proximity to the 5G network. In summary, we assume a simple 5G network where an application from a mobile end-host connects with another endpoint (fixed or mobile) over a low-latency path where the primary network bottleneck is the highly variable wireless 5G network link.

5.3.2.1 Rate Control Algorithm

The rate control algorithm, outlined in Algorithm 1, employs a delay-centric approach to maintain low network latency while preserving throughput stability. At its core, the algorithm dynamically adjusts the congestion window (cwnd) by analyzing real-time Round-Trip Time (RTT) distributions through two primary components: a sliding window backlog of length N

Algorithm 1: Rate Control Algorithm

```
Input: Backlog backlog ← FIFO queue of length N
Input: Histogram histogram ← Array[Number of buckets X]
Input: Bucket size $B$ ← Size (ms), $\Delta_{max} \leftarrow 10$
for each new RTT measurement do
    backlog ← backlog + $[RTT]$                                   ▷ Update FIFO
    $\mu \leftarrow \frac{1}{N} \sum_{i=0}^{N} \text{backlog}[i]$                  ▷ Compute average
    $b \leftarrow \lfloor \mu / B \rfloor$                                      ▷ Determine bucket
    $\text{histogram}[b] \leftarrow \text{histogram}[b] + 1$
    $\alpha \leftarrow \frac{\sum_{i=0}^{b} \text{histogram}[i]}{\sum_{j=0}^{X} \text{histogram}[j]}$
    if $b < X/2$ then
        $cwnd \leftarrow cwnd + (\alpha \times (X/2 - b) \times \Delta_{max})$
    else
        $cwnd \leftarrow cwnd - (\alpha \times (b - X/2 - 1) \times \Delta_{max})$
    end
end
```

and a histogram-based delay classification system that converges towards the desired RTT, based on the number of buckets X. For each incoming RTT measurement, the algorithm updates a fixed-length FIFO queue (backlog) of N recent RTT values, computes their moving average, and maps this average to a histogram bucket representing discrete latency ranges (e.g., 15 ms intervals). The histogram tracks the frequency distribution of these buckets over time, enabling the calculation of α—a normalized metric reflecting the cumulative probability of observing RTTs in the current or lower-latency buckets. Based on this probability and the bucket index, Hera modulates the congestion window asymmetrically: when operating in low-latency regimes (buckets below a predefined threshold), it increases the window proportionally to both α and the distance from the threshold, fostering aggressive utilization of available bandwidth. Conversely, in high-latency states, it reduces the window based on the severity of the observed delay, effectively curbing queue buildup. The algorithm incorporates safeguards to clamp the congestion window within empirically validated bounds, preventing extreme oscillations. In simple terms, the maximum change in sending rate occurs when a large number of recently observed RTTs are significantly different from previous observations, and these RTTs fall in the furthest buckets from the cen-

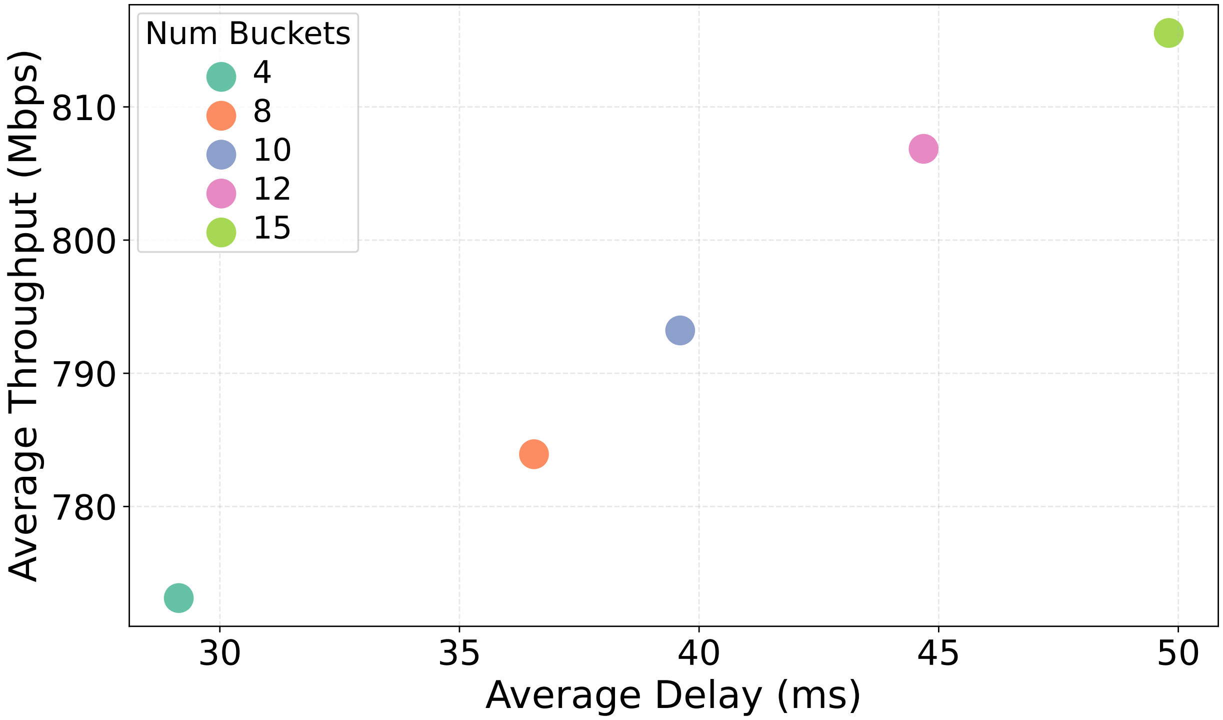


Figure 5.4: Impact of varying the number of buckets on protocol performance, measured in average throughput and delay. Results show that increasing the bucket count raises throughput but also increases delay.

ter. This dual mechanism—probabilistic delay classification coupled with gradient-based window adaptation enables Hera to preemptively mitigate congestion before packet loss occurs, making it particularly effective for latency-sensitive applications such as real-time video streaming, cloud gaming, and MIR applications where stable, low-delay communication is critical.

5.3.2.2 Hera rate control parameters

Hera requires four control parameters that drive its performance, i.e., Bucket Size, the Max Delta value, Histogram Limit, and the *backlog length*. These parameters can be finely tuned concerning the channel environment for enhanced performance gains.

Number of Buckets

The number of Buckets in the histogram determines the target RTT for the protocol. For a stable network environment, a low number of buckets is required, and for highly fluctuating network environments, a high number of buckets would be more suitable. For the general case, this value needs to be dynamically modified based on the network environment. For the VR streaming over 5G environments case, we can calculate a reasonable bucket size based on the RTT requirements of the VR applications for a smooth VR streaming experience. We can see the effects of changing the number of buckets in figure 5.4. As expected, the algorithm modifies the congestion window to ensure that the RTT converges towards the target RTT at the center of the distribution. For applications that prioritize higher throughputs with tolerance for higher RTTs, this parameter can be tuned to a higher value accordingly.

Max Delta

The maximum delta determines the maximum value by which the *cwnd* is modified at every RTT. It is used in combination with the current state and the alpha value, i.e., the probability of being in the current RTT or lower, to ensure that when the RTT is too high or too low, the cwnd is adjusted rapidly to converge towards the desired RTT. When the current state is close to the desired RTT, the protocol makes small adjustments to the cwnd. Increasing this value can cause the protocol to become unstable, as it may overreact to a signal and then change too rapidly in the opposite direction when attempting to recover and stay in this unstable loop.

Histogram Limit

The histogram limit is the maximum amount of data stored in the histogram, which is not only required for memory usage purposes but also to ensure that the global state of the network is not dependent on outdated data, which is not representative of the current network environment, as the state of the network environment can vary over time, especially for cellular networks.

Backlog length

The backlog stores a "recent history" of RTTs (Round-Trip Times). It stores the last few RTT measurements (e.g., the last 10 samples) in a sliding window. This helps smooth out temporary spikes or dips in delay. For example, if one RTT is unusually high (e.g., due to a random network hiccup), the backlog averages it with other recent RTTs to avoid overreacting. The backlog provides a short-term view of the network's current state, which is crucial for making quick, adaptive decisions. Increasing the backlog length would make the protocol slow to react to changes in the network environment, while decreasing it would make the protocol overreact to any observations.

5.4 Evaluation Methodology

5.4.1 VR Application

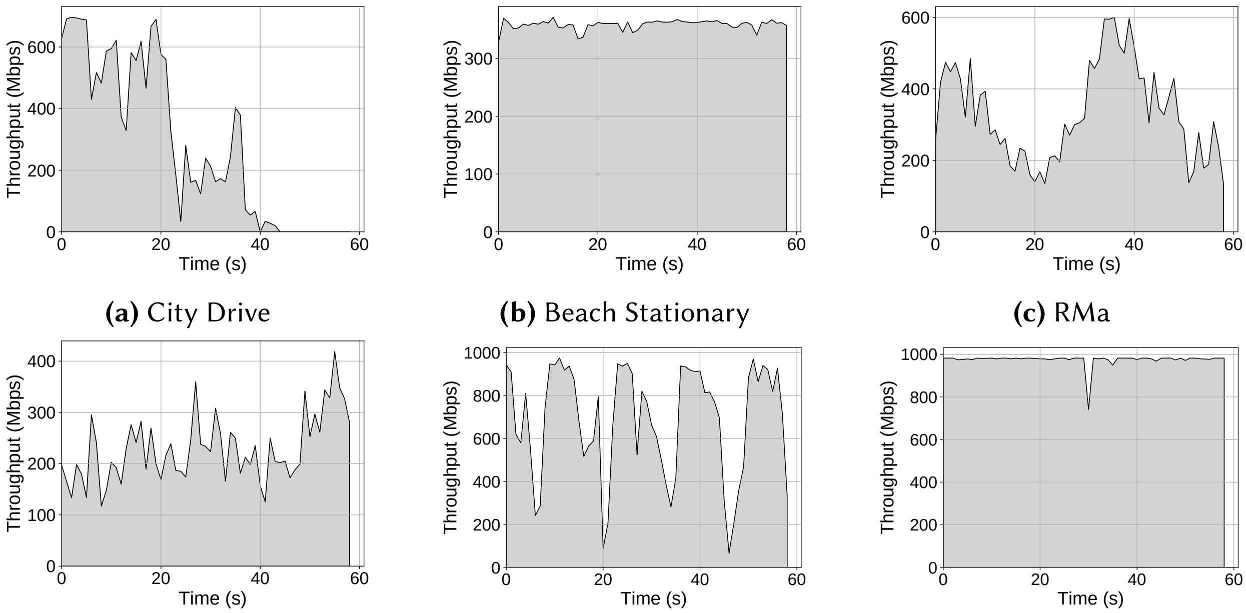
In order to compare the MIR application performance of different congestion control protocols, we develop a system for streaming adaptive video to a VR headset and measuring Quality of Experience (QoE). The system is built around three core components: a Linux server hosting DASH content via NGINX, a VR client using dash.js for playback, and dynamic switching of congestion control protocols.

The server is configured to host DASH streams using NGINX with an RTMP module, which hosts video content as chunks (e.g., 4–10 seconds) along with a Media Presentation Description (MPD) file for adaptive bitrate streaming. Videos are pre-processed into multiple bitrate representations (0.045–4 Mbps) using FFmpeg and MP4Box to ensure DASH compatibility. To test different congestion control protocols, the Linux kernel's TCP stack is adjusted using sysctl commands (e.g., switching between CUBIC and BBR). Experiments involve streaming a 10-minute video (e.g., Big Buck Bunny) to the VR headset under controlled network conditions with five clients streaming the video simultaneously.

On the client side, five clients run the stream using a modified dash.js player. The player fetches the MPD file and adaptively selects video segments based on real-time network conditions. The dash.js player logs client-side Quality of Experience (QoE) metrics, including frame losses, bitrate, and video quality, to gauge application performance from the user's perspective. This setup allows us to systematically compare five different TCP congestion control protocols, including Hera, to assess how each protocol influences the performance of the VR application in terms of network efficiency and user experience. In order to emulate the variable latency, packet losses, and capacity limits that may be observed in real network environments, we use the Linux Traffic Control (tc) and NetEm tools. This approach provides valuable insights into the impact of different network protocols on a VR application, offering a practical foundation for optimizing VR experiences in varied network conditions.

5.4.2 Network Trace Collection

In order to emulate a diverse set of realistic 5G network environments, we collect a set of channel traces that collectively cover a wide range of scenarios in terms of available bandwidth, mobility (driving, walking, and stationary), and variability. We employ different methodologies to generate this set of channel traces. Firstly, similar to prior work [257, 266], we use a real 5G connection from a commercial deployment in the wild, along with an Android application capable of recording the maximum available channel capacity by connecting to a remote server hosting dummy content specifically for this channel trace collection. These traces include urban driving and standing stationary at a beach, and the full channel traces are depicted in Figure 5.5(a) and Figure 5.5(b). Secondly, we use the mmWave module [166] built atop the NS-3 [205] tool developed by New York University wireless group (NYU Wireless) to generate channel traces emulat-



(a) City Drive **(b)** Beach Stationary **(c)** RMa

(d) UMa **(e)** Indoor Walking **(f)** Indoor Stationary

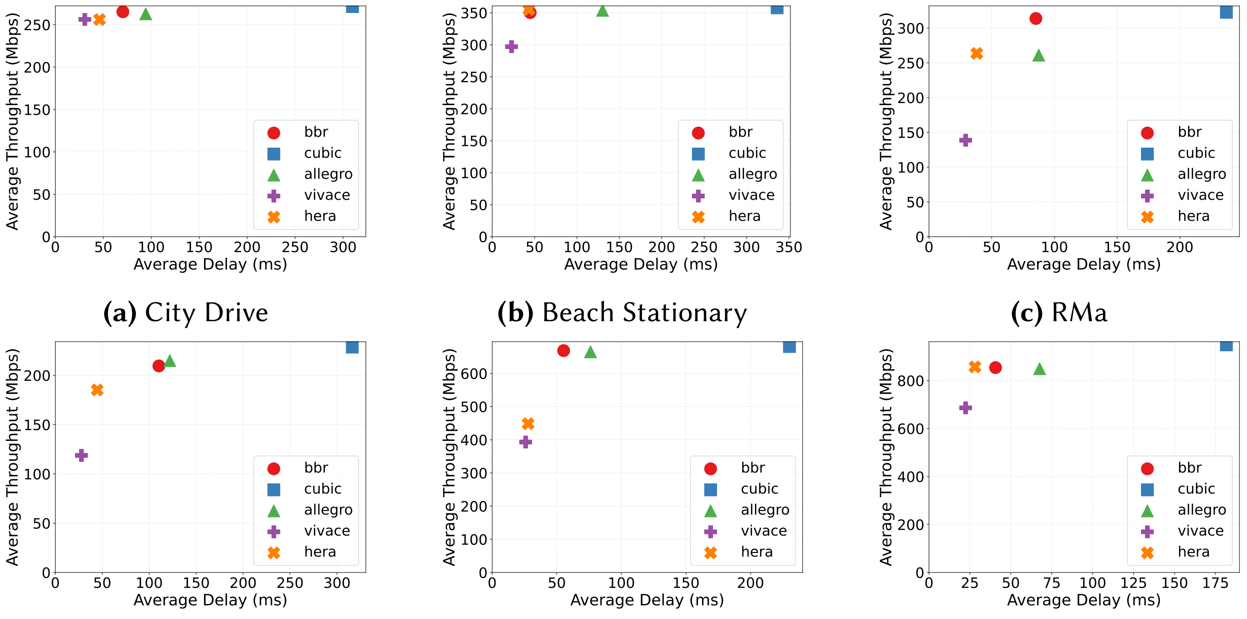
Figure 5.5: 5G network traces used for experiments were collected using a commercial 5G deployment, NS-3 mmWave simulations, and WiGig-based indoor setups, covering conditions MIR applications may encounter in real-world deployments. The traces include urban driving (City Drive, UMa), stationary scenarios (Beach Stationary, Indoor Stationary), rural mobility (RMa), and high-mobility indoor environments (Indoor Walking).

ing scenarios involving random user movement in an urban scenario, depicted in Figure 5.5(d), and a rural environment, depicted in Figure 5.5(c) using a spatial channel model [271]. Finally, we use channel traces generated using a high-speed WiGig router in an indoor environment in scenarios with high user mobility, depicted in Figure 5.5(e), and in a stationary scenario, shown in Figure 5.5(f). Altogether, these channel traces encompass a broad range of realistic network scenarios that a user of a MIR application may encounter in the real world.

5.4.3 Emulating the 5G traces

We used the Mahimahi framework [173] to emulate 5G network links based on the collected traces. Mahimahi's linkshell acts as a controlled router that queues packets and throws them at the desired rate as dictated by the trace file. The traces give us the ability to run several algorithms and compare the performance across scenarios in a unified and controlled manner without having to control for many of the external parameters that are usually faced by running things in the wild, i.e., unpredictable competing traffic from other users or random obstacles and user movements in the environment. A similar approach has been used by Verus [266] and Sprout [257] in the past.

We have mimicked the popular *iperf* utility for our experiments, having separate sender and receiver programs. The receiver runs inside a Mahimahi linkshell, while the sender repeatedly transmits fixed-size blocks of 128 KiB to the receiver for 60 seconds. We chose $bucket\_size = 15$, $backlog\_length = 10$ and the $Max\_delta = 10$, as our default parameters for Hera. We choose 4 prominent CC protocols for comparison in our experiments. These protocols include BBR, Allegro, Vivace, and Cubic.



(a) City Drive **(b)** Beach Stationary **(c)** RMa

(d) UMa **(e)** Walking **(f)** Stationary

Figure 5.6: Comparison of congestion control protocols over six 5G network traces, evaluating their average throughput (y-axis) and delay (x-axis). BBR and Cubic achieve high throughput but at the cost of increased latency. PCC Vivace maintains lower delays but struggles under highly fluctuating conditions. PCC Allegro exhibits inconsistent performance, sometimes achieving low delays but often suffering from high latency similar to legacy TCP protocols. Hera achieves a balanced performance, maintaining comparable throughput to BBR and Cubic while significantly reducing delay—up to 50% lower than BBR and 66% lower than Cubic's delay in some cases.

5.4.4 Hera Rate Control Implementation

We implemented the Hera QoE-aware rate control module in the Linux kernel as a pluggable congestion control module [67, 108]. The protocol has been tested on Ubuntu 20.04.1 with kernel version 5.15.0-41.

5.5 Hera Evaluation

We aim to compare Hera's performance to other state-of-the-art protocols from several aspects, i.e., the VR streaming performance of the protocol in terms of various VR streaming QoE metrics as well as a traditional throughput-over-delay performance comparison in an emulation environment utilizing the collected 5G traces.

5.5.1 Baseline Selection

We evaluate our congestion control algorithm against four representative baselines that span the spectrum of established and emerging approaches in the context of AR/VR streaming. These include loss-based, model-based, learning-based, and AR/VR-specific schemes, each reflecting different trade-offs between throughput, latency, and stability.

- **Cubic:** The default congestion control algorithm in most operating systems, Cubic represents the class of loss-based controllers. While not optimized for low-latency media, it establishes a widely deployed baseline for throughput and fairness.

- **BBR:** BBR is a modern congestion control algorithm that uses bottleneck bandwidth and minimum RTT estimation to optimize throughput with controlled delay. Its proactive model-based design makes it well-suited for real-time interactive applications.

- **PCC Allegro:** Developed specifically for immersive media streaming, Allegro offers a domain-aware baseline that prioritizes frame delivery deadlines and low buffering, directly aligning with the requirements of AR/VR systems.

- **PCC Vivace:** A reinforcement learning-based algorithm that adapts to dynamic network conditions using latency and throughput feedback. It serves as a strong representative of learned congestion control strategies.

We do not include traditional delay-based controllers such as Vegas [40], LEDBAT [221], TIMELY [167], or Copa [24]. Vegas and LEDBAT are known to be overly conservative and underperform in shared environments, especially when competing with aggressive loss-based flows like Cubic [50, 144]. TIMELY requires hardware timestamping, which is generally unavailable in wireless and edge-based AR/VR deployments. Copa, while theoretically promising, is not widely adopted and lacks open-source support for immersive media use cases. Additionally, delay awareness is already represented in our chosen baselines—BBR and Vivace both integrate delay feedback using more robust and adaptive mechanisms. Therefore, we focus on practically relevant and competitive baselines tailored to our target domain.

5.5.2 Emulated 5G Traces Evaluation

In this subsection, we compare Hera's performance to four other congestion control protocols: Google's BBR, PCC Allegro, PCC Vivace, and the de facto legacy TCP Cubic. We ran each of these protocols across six measured 5G channel traces using MahiMahi. Each protocol was run five times using different seeds to be able to get enough statistical rigor in the results, as well as being able to compute the statistical significance of both the achieved throughput and delay.

Figure 5.6 presents the results comparing Hera against the four other chosen congestion control protocols over the emulated 5G traces. For the scatter sub-figures, we plot one point per congestion control protocol, corresponding to its measured average throughput (y-axis) and delay (x-axis) combination, averaged over five different seed runs per protocol. A protocol that strongly prioritizes ultra-low latency at the cost of low throughput exists in the bottom-left quadrant of this graph. As the protocol behavior trends towards prioritizing throughput and "aggressively" increasing the sending rate, the protocol will gradually move from the bottom-left quadrant towards the top-left quadrant until the protocol starts sending data beyond the available channel capacity, causing bufferbloat-induced queueing delays, which will cause the protocol to move towards the top-right of this graph. In other words, the optimization goal of a protocol determines its place on this curve. For most applications, the ideal operating point on this curve is in the top-left quadrant, only slightly below the area where the curve starts moving towards the top-right, which indicates an increase in latency. This is especially true for MIR applications that demand high bandwidth and also ultra-low latency in order to maintain a smooth and high-quality QoE. Several key takeaways can be seen across the scatter comparison results of the six traces.

Legacy congestion control protocols tends to have the highest average throughput compared to the rest of the protocols. However, this throughput comes at the expense of much higher delays, often achieving more than three times higher delays compared to other protocols.

Google's BBR achieves a similar average throughput to TCP Cubic while lowering the delay significantly compared to Cubic. In all experiments, BBR hovers around the ideal top left quadrant. In all of the experiments, Hera is able to achieve lower delays than BBR except for two cases where the delay performance is almost identical. This delay performance improvement of up to 50% in certain cases comes at a cost of a less than 10% reduction in throughput performance when compared against BBR.

PCC Vivace generally achieves low delays except in the two most highly fluctuating traces, City Driving and WiGig Walking, which cause Vivace to suffer large delays due to it being unable to adapt to the high fluctuations. In two scenarios, Vivace is able to outperform both Hera and BBR in terms of delay performance for up to a 30% decrease in throughput utilization.

PCC Allegro has an inconsistent performance, achieving relatively low throughput and delays in certain traces. In the general case, it tends to achieve high throughput but with high delays as well, similar to the legacy congestion control protocols Cubic in half of the tested scenarios.

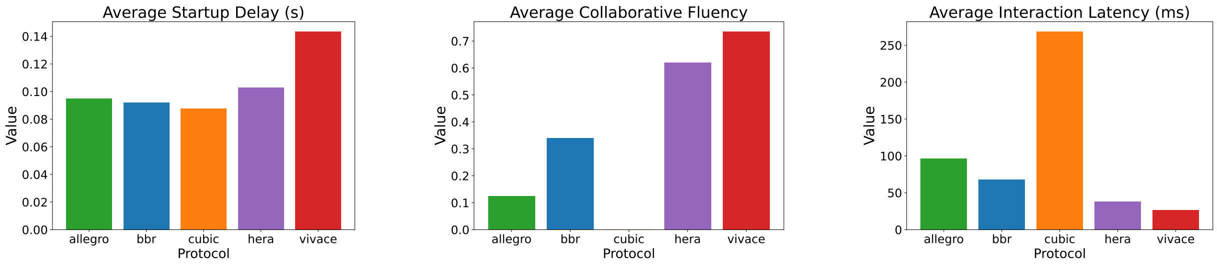
Finally, Hera achieves a good balance across the different traces, achieving almost the same throughput across the six traces as BBR and Cubic while maintaining lower delays, about half of that of BBR in some cases, such as UMa, RMa, and most notably in the Beach Stationary trace, where Hera also manages to outperform other protocols in throughput while still maintaining lower delays. This ability to maintain high throughputs while achieving lower delays enables Hera to outperform the other congestion control protocols in terms of VR streaming performance, as we observe in Section 5.5.3.

5.5.2.1 Observable QoE Implications

While the preceding analysis focuses on throughput and latency, these network metrics translate directly into observable Quality of Experience (QoE) characteristics for users in collaborative XR applications. Table 5.1 summarizes key QoE metrics that can be inferred from throughput and latency measurements and how they manifest in user experience. In our experiments, the different congestion control protocols display varying throughput-latency tradeoffs that map directly to observable QoE outcomes in collaborative XR applications:

Cubic: While Cubic achieves high average throughput, it does so at the cost of substantial latency increases, often exceeding acceptable limits for interactive MIR applications. As a result, users would experience frequent delays in seeing their collaborators' actions reflected in the shared space, leading to impaired collaborative fluency and responsiveness. The high latency also increases motion-to-photon delays and may contribute to discomfort or disorientation.

BBR: BBR generally balances high throughput with lower latency than Cubic, enabling better QoE. Users are likely to experience smoother interactions and fewer stalls, with video resolution often maintained at higher levels. However, under certain highly variable network conditions, BBR may still cause latency spikes, resulting in momentary interaction lag.



(a) Average Startup Delay (s)

(b) Average Collaborative Fluency

(c) Average Interaction Latency (ms)

Figure 5.7: Comparison of QoE metrics across protocols: startup delay, collaborative fluency, and interaction latency.

PCC Allegro: Allegro shows inconsistent throughput and latency behavior across traces. In scenarios where it behaves aggressively, it may cause similar QoE degradation to Cubic, such as higher stalls and lower collaborative fluency. In other cases, its lower throughput would force the system to reduce video resolution, leading to a loss of visual fidelity without significant latency gains.

PCC Vivace: Vivace is typically able to maintain lower latency, resulting in good responsiveness and collaborative fluency in stable conditions. However, in highly fluctuating environments (e.g., City Drive, Indoor Walking traces), Vivace struggles to adapt, leading to sharp QoE degradation through either increased buffering or reduced resolution to compensate for throughput drops.

Hera: Hera consistently achieves low latency while maintaining high throughput across diverse network conditions. This directly supports superior QoE: fast startup times, minimal stall events, sustained high video resolution (e.g., 4K where possible), and fluid, responsive collaborative interactions. Hera's design prioritizes latency stability, making it particularly effective at preserving immersion and reducing discomfort in XR environments.

To visualize this mapping, Figure 5.7 summarizes the relative QoE performance of the tested protocols across key metrics, including startup delay, resolution, and interaction responsiveness, derived from their throughput-latency characteristics in our experiments.

5.5.3 VR Multi-User Application

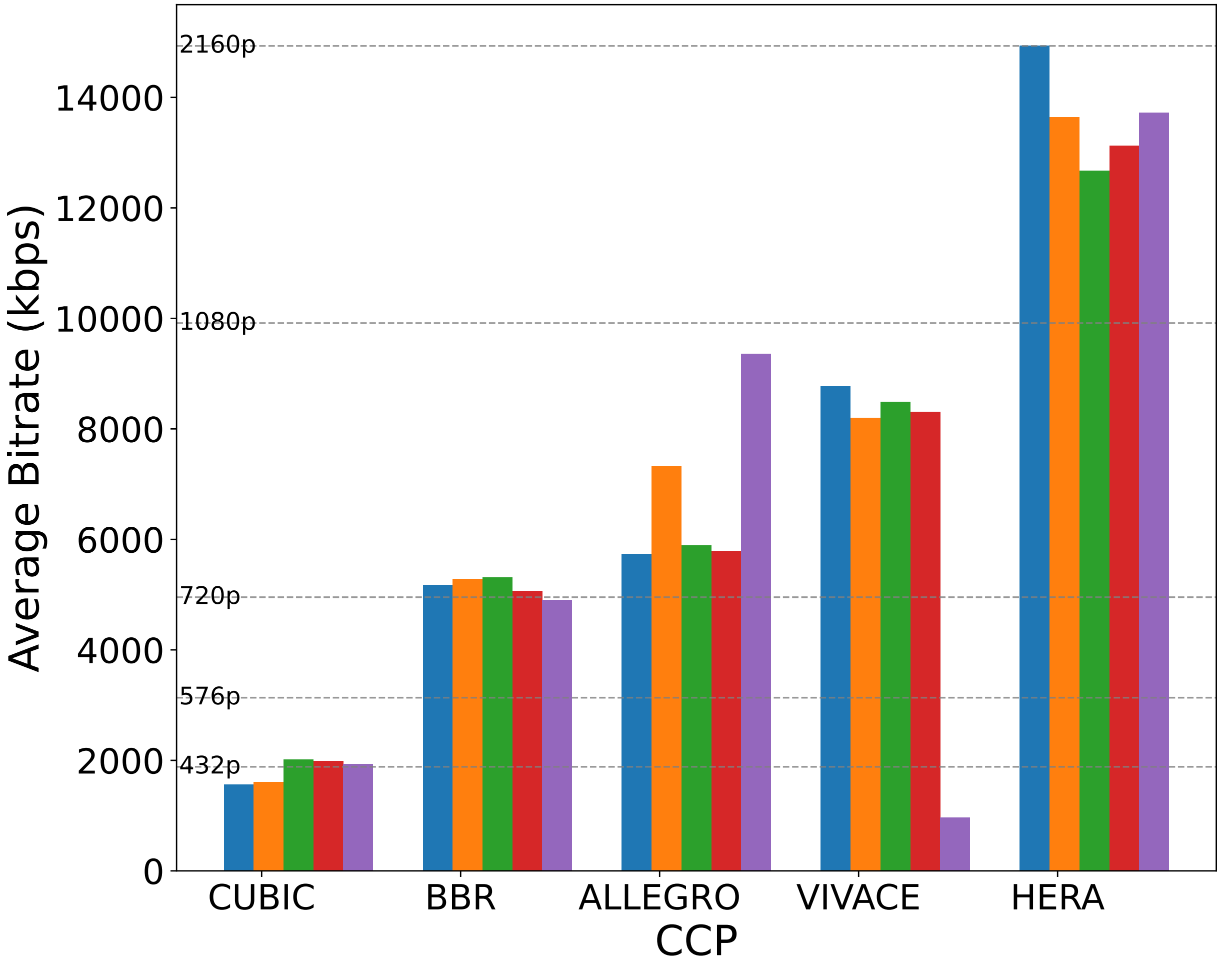


Figure 5.8: Average bitrate of protocols in Mixed Reality streaming applications implemented on Meta Quest headsets. Five colors represent five simultaneous streams. Hera maintains 4K quality, while BBR and Cubic degrade to lower resolutions due to congestion. Allegro and Vivace perform better but show fairness issues, leading to inconsistent QoE.

We evaluate the performance of five congestion control protocols when supporting a real Mixed Reality application with users running the application on a range of devices, including Meta Quest headsets. Figure 5.8 shows the average bitrate, which corresponds to the video playback resolution, of five simultaneous streams in a realistic network environment with different congestion control protocols supporting the server. As the videos are encoded in multiple levels of quality, when the system detects that the video buffering rate is low due to congestion, the Adaptive Bitrate Algorithm reduces the video quality. The results show Hera outperforming existing protocols and consistently maintaining 4K stream quality for all clients, while BBR drops to 720p stream quality and Cubic performs even worse due to packet losses in the network environment. Allegro and Vivace perform slightly better for some users, but we observe that both protocols have outliers, which indicates fairness issues causing the QoE for different users to vary. The experiments in Figure 5.6 illustrate that Hera consistently achieves lower throughput than some protocols. Despite this, we observe higher streaming QoE for Hera due to the fact that higher latencies incurred by aggressive protocols directly degrade streaming quality by introducing delays in the delivery of video and audio data, causing buffering and consequently causing the ABR algorithm to lower the stream quality. Although our analysis also revealed minor frame drops across all protocols, with Hera exhibiting the lowest and Cubic the highest frequency, these differences did not translate to a noticeable impact on the user experience, as the system dynamically adapts the video quality to ensure minimum interruptions at the cost of video resolution.

5.5.4 Fairness Analysis

A critical component of any congestion control protocol is the ability to operate in an environment with multiple network flows using different congestion control protocols . This is especially true in MIR applications where multiple users simultaneously use the application and compete for network resources. As such, we evaluate the Jain fairness index [126] for all congestion control protocols for traditional as well as MIR applications. As discussed in Section 5.5.3 where we observe the performance of different protocols supporting five competing clients accessing the application content simultaneously. We observe that Cubic and BBR display strong intra-fairness, while PCC Allegro and Vivace show outliers that lower the fairness score. Hera demonstrates slightly lower fairness than Cubic and BBR but with significantly improved video quality for each user.

Additionally, we conduct various experiments using our experimental framework to analyze the intra-fairness behavior of the selected congestion control protocols when operating in a 5G network environment to replicate the scenario where multiple users in a local environment are simultaneously accessing the application and competing for network resources in a 5G network environment. For traditional scenarios, as discussed in Section 5.5.2, we evaluate the behavior of each congestion control protocols when operating in the *Beach Stationary* simulated 5G network environment using the network trace collected from a real 5G deployment in the wild. The results of these experiments are illustrated in Figure 5.9. For our experiments, we run 6 competing flows and observe the fairness characteristics of our protocol in terms of throughput allocated over time to each flow. We also employ the widely used Jain's fairness index [126] metric to quantify the overall fairness characteristics of each congestion control protocols in these experiments. The results corroborate the findings from the MIR streaming experiments. In our experiments, Cubic, BBR, and Hera demonstrate strong fairness characteristics with above 90% Jain's fairness, allocating an equal share of the available bandwidth across all connected clients, with Allegro and Vivace falling behind due to a small subset of the flows dominating the other flows in the case of Allegro, or performing significantly worse in terms of throughput compared to the other flows in the case of Vivace.

5.6 Discussion

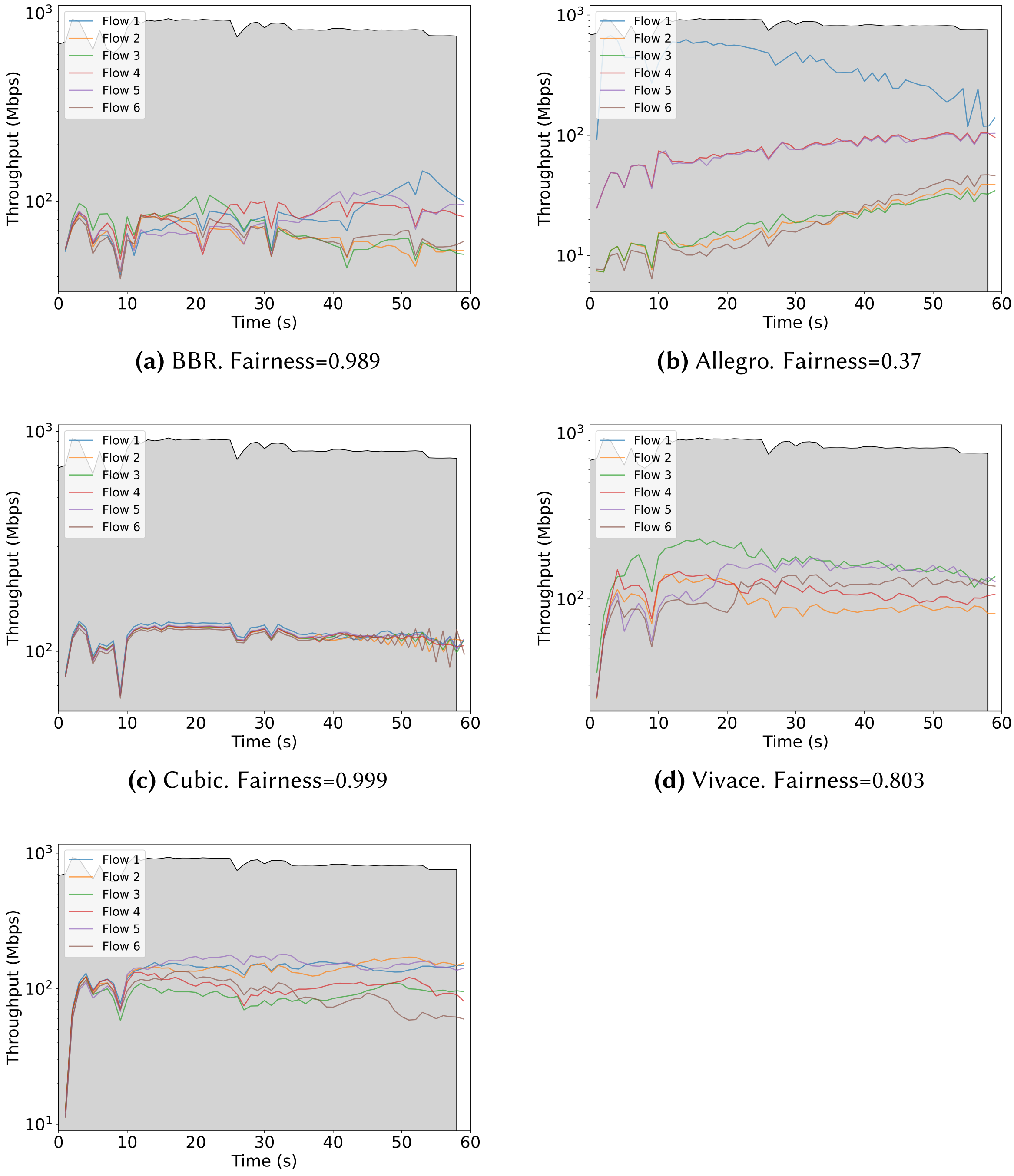
Performance Discussion

Traditional congestion control algorithms exhibit several fundamental weaknesses when applied to MIR streaming over 5G networks, leading to suboptimal performance compared to Hera. Conventional loss-based TCP algorithms such as Cubic and Allegro assume that packet loss signals network congestion. However, in cellular networks, packet losses frequently occur due to handoffs, signal fluctuations, and interference, rather than genuine congestion. This results in unnecessary rate reductions, degrading throughput and streaming quality. Also, the end-to-end nature of TCP means that congestion response happens far from the point of signal fluctuation, leading to delayed adaptation [159]. In highly dynamic 5G environments, where channel conditions fluctuate on millisecond timescales, algorithms like BBR struggle to converge because they rely on multi-RTT bandwidth estimation windows (6-10 RTTs) [46]. This mismatch results in slow reactions to rapid network changes, leading to both underutilization and excessive queuing.

Hera overcomes these limitations by employing a histogram-based RTT tracking mechanism to dynamically adjust its congestion window (cwnd), allowing it to proactively respond to latency fluctuations rather than relying on delayed congestion signals. Unlike loss-based algorithms, Hera avoids unnecessary rate reductions due to non-congestive packet losses, which are common in 5G environments. By prioritizing low-latency operation while maintaining stable throughput, Hera prevents both over-congestion, as seen in BBR, and under-utilization, which affects protocols like Allegro and Vivace in highly variable networks. These optimizations allow Hera to deliver consistent high video quality with minimal delay, making it particularly effective for bandwidth-intensive, latency-sensitive MIR applications.

5.7 Conclusion

This chapter presents Hera, a modular framework that integrates a novel QoE-aware rate control protocol with a high-level AR/VR application layer designed to meet the demanding requirements of multi-user immersive reality experiences over next-generation wireless networks. Through extensive evaluation using realistic 5G network scenarios and a custom-built streaming and benchmarking system, we demonstrate that Hera consistently achieves lower latency, higher throughput, and greater fairness compared to state-of-the-art congestion control protocols including BBR, Allegro, Vivace, and Cubic. By combining the low-level congestion control module with an application-aware streaming and synchronization layer, Hera enables dynamic adaptation of video resolution, frame rate, and collaborative update rates in response to real-time network conditions. Our results show that this integrated design significantly improves key Quality of Experience (QoE) metrics such as startup delay, stall frequency, and responsiveness that are critical to sustaining high-quality multi-user AR/VR sessions. Overall, this work illustrates the requirements to unlock the full potential of immersive multi-user experiences on emerging 5G and future wireless infrastructures. Future work will explore scaling the framework to larger user groups, incorporating edge computing support, and extending the congestion control techniques to accommodate new transport protocols and network architectures.



(a) BBR. Fairness=0.989

(b) Allegro. Fairness=0.37

(c) Cubic. Fairness=0.999

(d) Vivace. Fairness=0.803

(e) Hera. Fairness=0.965

Figure 5.9: Throughput distribution across six competing network flows for different congestion control protocols in a 5G network environment. Cubic, BBR, and Hera achieve high fairness (Jain's fairness index > 0.96) by evenly distributing available bandwidth. Allegro and Vivace, however, show poor fairness, with Allegro exhibiting extreme disparities where certain flows dominate bandwidth, and Vivace struggling with significantly lower throughput across flows.

6 | Are AI Services Going Green?

ABSTRACT

Current carbon offsetting strategies are failing to mitigate the carbon footprint of AI data centers at an environmentally sustainable pace. The rapid growth of AI services in emerging regions is exponentially amplifying computational demand in data centers, leading to significant energy consumption and a substantial carbon footprint. To mitigate their environmental impact, operators increasingly rely on carbon offsetting strategies including carbon credits, Renewable Energy Certificates (RECs), and Virtual Power Purchase Agreements (VPPAs). Unfortunately, the current implementations of these strategies by data centers do not sufficiently reduce the carbon footprint produced by the flood of AI services entering the market. This chapter critically evaluates the effectiveness of these prevailing strategies in reducing the carbon footprint of energy-intensive AI data centers. We identify key implementation challenges, create a sustainability calculator to model the potential emissions generated by a system, and propose targeted solutions to enhance accountability, ensure meaningful emissions reductions, and prevent a potential climate crisis.

6.1 Introduction

Recent global efforts to transition from energy-intensive computational services to a sustainable energy model can not keep up with the pace at which new AI services are flooding the market.

As the world hangs on the precipice of an irreversible climate disaster [206], the rapid growth of energy-intensive AI services being developed and deployed at an unprecedented rate may tip the scales in the wrong direction. In emerging regions, multiple challenges exist that make the goal of environmentally sustainable AI services an uphill battle. Renewable energy production in emerging regions is limited and insufficient for meeting the current energy demands of emerging regions; the additional burden of data centers powering new energy-intensive AI services increases the energy demand exponentially with no renewable energy sources available to offset the increased energy requirements. To mitigate these issues, the energy requirements for AI services and the complexity of AI models used by these services in emerging regions needs to be minimized.

Training state-of-the-art AI models consumes many MWH of electricity, often sourced from carbon-intensive grids, resulting in a substantial and growing carbon footprint. The International Energy Agency (IEA) predicts that AI-related energy demand could increase by more than double by 2030 [1], raising concerns that unchecked growth could offset progress toward global sustainability goals. The tech industry, once a leader in commitments to net-zero emissions, now faces significant setbacks due to AI's escalating energy needs. Companies have adopted various strategies to mitigate their environmental impact, including carbon credits, investments in renewable energy, and nuclear power. However, many approaches fall short of addressing the root problem. Significant effort and investments have been made in recent times pushed by governments, organizations, companies, and individuals to reduce carbon emissions and mitigate the environmental damage that threatens to make the world uninhabitable. Prominent strategies that have been implemented to reduce the carbon footprint of these services include carbon credits, Purchase Power Agreements, and carbon taxes imposed by regulatory authorities. Table 6.1 summarizes existing carbon offset mechanisms and their limitations.

In this chapter, we analyze the limitations of existing strategies in decarbonizing AI infrastructure, highlighting structural challenges that render them inadequate as a primary sustainability

Carbon Offset Strategy	**Description**	**Limitations**
Carbon Credits	Financial instruments representing reduction or removal of one tonne of emissions through environmental projects (e.g., reforestation, renewable energy).	• Additionality concerns • Permanence risks • Verification challenges • Leakage
RECs	Tradable certificates proving 1 MWh of electricity was generated from renewable sources and fed into the grid.	• Unbundled RECs • Location mismatch • Time decoupling • No grid decarbonization
VPPAs	Financial contracts where buyers purchase renewable energy and RECs from specific projects at fixed prices, with energy delivered virtually through the grid.	• Physical disconnect • Complex implementation • Price volatility risk • Grid congestion • Long lead times

Table 6.1: Common Carbon Offsetting Strategies for AI Data Centers

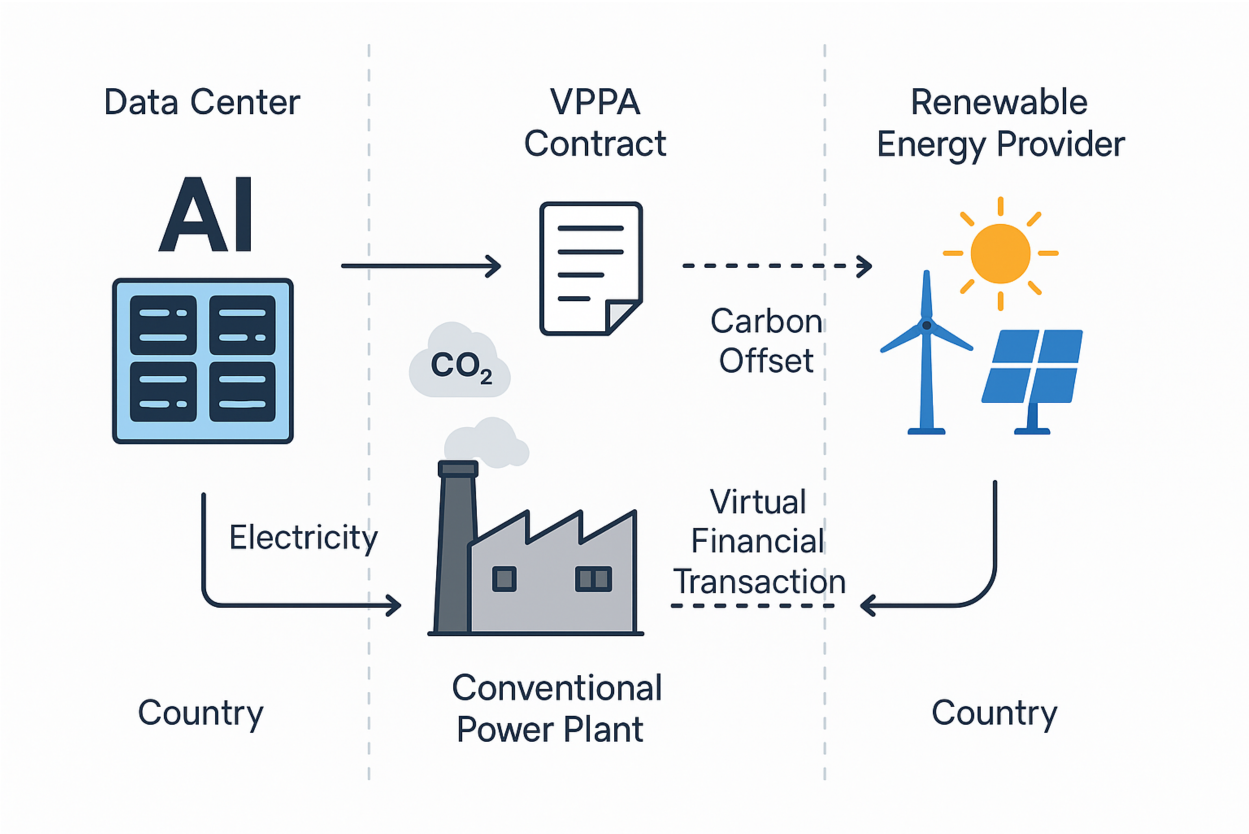


Figure 6.1: Virtual Power Purchase Agreement

solution. We further discuss alternative pathways such as geographically constrained renewable procurement, advanced energy efficiency measures, and policy-driven grid decarbonization that could more effectively align AI's growth with climate objectives. Finally, we present a *Sustainability Calculator* to evaluate the environmental sustainability of a system. Our findings underscore the urgency of rethinking carbon accounting and energy sourcing practices to prevent AI from becoming a dominant driver of global emissions in the coming decade.

6.2 Background and Challenges

The rapid proliferation of AI workloads has led to exponential increases in energy consumption, creating a growing urgency to ensure that AI infrastructure is environmentally sustainable. A significant body of research has emerged in recent years to assess the carbon footprint of AI, improve computational efficiency, and evaluate the effectiveness of current decarbonization strategies. Prevailing carbon offsetting mechanisms such as VPPAs, RECs, and carbon credits face multiple challenges that undermine their effectiveness.

Geographic and Jurisdictional Misalignment. Several studies have pointed out the flaws in current offsetting mechanisms due to mismatches between where renewable energy is generated and where it is consumed. This decoupling impairs the ability of regulatory bodies to validate emissions reductions. For example, many VPPAs are signed between companies in the U.S. and renewable providers in regions with abundant renewable energy like Texas, regardless of where the energy is actually consumed [110]. A 2024 study in *Nature Energy* found that 43% of cross-border VPPAs resulted in net emissions increases due to geographic mismatches and transmission inefficiencies. These misalignments erode accountability and inflate sustainability claims.

Scalability Limits of Offsetting Strategies The scalability of current offsetting solutions is significantly outpaced by the growth of AI workloads. While the AI industry is projected to grow its energy consumption by 30 to 50% annually [122], VPPA capacity is expanding at only 8 to 15% per year. This widening gap means that even with growing investment in offsets, a decreasing proportion of total emissions will be covered. Amazon's renewable energy targets, for instance, require 35 GW of new VPPAs by 2030, equivalent to 10% of the entire U.S. solar capacity in 2023, posing enormous feasibility challenges given current transmission and permitting bottlenecks.

Additionality and Permanence Concerns A major concern with carbon credits and VPPAs is the lack of additionality, whether projects funded through offsets would have occurred anyway. Henderson et al. [110] found that fewer than 15% of corporate VPPAs resulted in new renewable capacity. Further complicating matters, permanence is not guaranteed. Forest-based carbon credits, for instance, face the risk of reversal through deforestation and wildfire [79], introducing long-term uncertainty into carbon accounting.

Temporal and Locational Decoupling of RECs. RECs, though widely used, often fail to ensure that clean energy is used in real-time or in the same location as the claimed consumption. Dodge et al. [79] have shown that cloud carbon intensity varies significantly by region and hour, emphasizing that offset mechanisms such as RECs need to be aligned both geographically and temporally to effectively reduce actual emissions.

Market Distortion and Social Costs. Fixed-price VPPA contracts are designed to provide data centers with price certainty, but they often pass the rising cost of renewable energy to local consumers. These contracts guarantee large corporate buyers low prices over 10–20 years, even as market rates increase due to inflation or supply chain bottlenecks. This pricing imbalance can raise electricity costs for small businesses and households [240]. For example, assuming a conservative annual increase of 6% in market electricity prices, in line with historical trends from the U.S. Energy Information Administration [240], a market rate of $45/MWh in year 0 would rise to approximately $80/MWh by year 10. This results in a $35/MWh price gap between fixed VPPA rates and prevailing market rates, a discrepancy that utilities often recover by increasing rates for residential and small commercial users. As a result, fixed-price VPPAs may unintentionally drive up electricity costs for non-participating consumers, exacerbating energy affordability challenges and creating structural inequities in access to clean power.

Illusion of Sustainability. Offsetting strategies used in isolation can foster a false sense of progress toward sustainability. Many firms report carbon neutrality while continuing to rely on fossil-powered grids. As noted by Patterson et al. [189], lifecycle emissions for hyperscale AI deployments can exceed 500,000 tons of CO_2 when accounting for training, inference, and hardware manufacturing. Green AI initiatives [219, 265] have improved computational efficiency, but efficiency gains are often offset by the Jevons paradox [16]: better performance at lower cost encourages even more usage. As a result, relying solely on accounting mechanisms like VPPAs and RECs risks delaying the adoption of more systemic solutions such as on-site renewables, grid decarbonization, or advanced nuclear energy [92, 122].

Policy Gaps and Regulatory Bottlenecks. Finally, the absence of robust carbon accounting standards exacerbates the challenges above. While some policy progress is underway such as the EU's Carbon Border Adjustment Mechanism (CBAM) [83] and the U.S. Inflation Reduction Act [237] implementation remains uneven and fragmented. Cowls et al. [71] and Rae et al. [195] emphasize the need for binding emissions reporting and mandatory clean energy procurement for AI services. Without enforceable standards, voluntary offsets will continue to dominate, often with questionable environmental outcomes.

6.3 Case Studies

In this section, we will present three real world examples of carbon offsetting strategy implementations that were not able to complete the desired goals due to the challenges discussed previously.

6.3.1 Renewable Offsets vs. Real Emissions: Limits of VPPAs and RECs

Cloud AI operators often claim their data centers are powered by *100% renewable* energy through off-site virtual power purchase agreements (VPPA) or renewable energy certificates (REC). However, this claim is often misleading, as these mechanisms often do not account for the temporal and geographic mismatches between clean energy production and actual consumption in the data center. For example, an AI data center that operates around the clock can purchase RECs from a solar farm equal to its annual energy use but still draw power from coal-fired sources during the night when the solar farm is not producing. These unbundled RECs are inexpensive and have been criticized for greenwashing, as they do not necessarily promote new renewable energy generation or reduce the data center's reliance on fossil fuels. Industry experts emphasize that unbundled RECs cannot credibly support claims of 100% renewable energy use, particularly when they are sourced from distant projects.

To address this, tech companies such as Microsoft have begun exploring *24/7 carbon-free energy* strategies, aiming to match energy consumption with clean power on an hourly basis. Microsoft tested this approach at its Netherlands campus by contracting more wind and solar capacity than its average demand to strive for round-the-clock renewable energy supply. Despite these efforts, Microsoft's sustainability reports indicate continued reliance on conventional RECs, revealing the persistent challenge of aligning AI infrastructure's energy needs with carbon-free power at all hours. This gap between virtual offsets and actual emissions highlights the need for more robust solutions, such as on-site renewable generation, energy storage, or investments at the grid level, to ensure the sustainability of AI operations.

6.3.2 Data Center Growth vs. Grid Capacity

Ireland provides a clear example of the strain that AI infrastructure places on national energy systems. Data centres consumed 21% of Ireland's total metered electricity in 2023 [52], with projections indicating this share could approach one-third of national demand by 2026. This unsustainable growth triggered a de facto moratorium on new data center construction in Dublin until 2028. This dramatic increase in energy demand, driven by cloud and AI services, has outpaced the growth of local power generation, raising serious concerns about grid reliability and climate impact. In response, Ireland's grid operator imposed a de facto moratorium on new data center connections in the Dublin area, the epicenter of the country's tech infrastructure. This pause is expected to last until at least 2028. Local authorities have also become more cautious: in 2024, the South Dublin County Council denied planning permission for a proposed 72,000 m^2 Google data center, citing the project's failure to demonstrate a sustainable power source to support its immense electricity demand. Industry reports indicate that Ireland's "chronic" power capacity shortage is forcing companies to reconsider their expansion plans. Microsoft, for example, announced in 2025 that it would shift new data center investments away from Ireland to regions with more abundant green energy [247].

Despite these challenges, hyperscale cloud firms have invested heavily in Irish renewables, with Microsoft alone securing about 900 MW of wind and solar power purchase agreements (PPAs), which are expected to cover around 28% of Ireland's 2030 renewable energy target [169, 178]. However, these clean energy purchases cannot resolve the immediate grid constraints, prompting the consideration of new legislation that would allow data center operators to build and manage their own power infrastructure to meet growing demand. Ireland's experience illustrates the critical need to align AI data center growth with timely grid upgrades and local renewable energy generation. Without this alignment, even a country with a favorable technological environment can be forced to curb expansion due to sustainability and energy security concerns.

6.3.3 Carbon Cost of Training AI Models: GPT-3 vs. BLOOM

The training of modern AI models has revealed significant sustainability challenges, as illustrated by concrete case studies. One landmark example is OpenAI's GPT-3, a large language model with 175 billion parameters, which required approximately 1,287 MWh of electricity and emitted around 552 metric tons of CO_2e during its training process [**Patterson2021**, 154]. This is equivalent to the annual carbon footprint of over a hundred gasoline cars, just for the creation of a single AI model. Such numbers raise doubts about the effectiveness of carbon offsets, particularly when many models are trained in regions where the marginal power comes from fossil fuels. However, not all training processes are equal. The open-source BLOOM model, also with 176 billion parameters, was developed with an emphasis on energy efficiency and hosted partly on France's low-carbon grid. BLOOM's training required only about 433 MWh and emitted 30 t CO_2e, resulting in over a 90% reduction in emissions compared to GPT-3. This substantial reduction was achieved through a combination of cleaner energy, more efficient hardware (specialized AI accelerators), and algorithmic optimizations. In fact, research by Google suggests that by using state-of-the-art processors and running workloads in optimized, renewables-powered data centers, the carbon footprint of training a model of GPT-3's size can be reduced by a factor of 100 to 1000.

This comparison highlights that the location and method of AI model training play a critical role in determining sustainability outcomes. Moreover, the energy demand does not end at deployment: inference for generative AI services is rapidly becoming a significant load on data centers. Estimates show that each query to an AI chatbot consumes significantly more energy than a standard Google search query [74], due to the computational demands involved in generating responses. With millions of such queries being processed daily, the cumulative energy use (and associated emissions) from inference can quickly surpass that from training. This situation has important social and policy implications. On one hand, AI developers are increasingly factoring carbon costs into their design processes as part of “Green AI” initiatives, and some are publishing emissions reports for transparency. On the other hand, there is growing support for incentivizing or regulating AI training and inference to occur on grids with excess clean energy or during renewable generation peaks. Studies suggest that such approaches could reduce emissions by an order of magnitude or more.

6.4 Sustainability Calculator Draft

6.4.1 Motivation and Scope

As artificial intelligence (AI) systems grow increasingly complex and ubiquitous, their environmental footprint becomes more difficult to account for using coarse approximations or single-factor models. Existing sustainability assessments often restrict their analysis to operational emissions, neglecting the carbon costs embedded in hardware production, infrastructure overheads, and dynamic inference workloads. We propose a holistic Sustainability Calculator that integrates the full AI service pipeline—from hardware manufacturing and infrastructure maintenance to training and real-time inference—thereby enabling a more robust and actionable estimation of total carbon emissions.

6.4.2 Model Formulation

We define the total lifecycle carbon emissions of an AI service, C_{total}, as the sum of four principal components: emissions due to hardware manufacturing (C_{hw}), data center infrastructure overhead (C_{infra}), model training energy (C_{train}), and emissions incurred during inference (C_{infer}). Formally, we express:

$$C_{\text{total}} = C_{\text{hw}} + C_{\text{infra}} + C_{\text{train}} + C_{\text{infer}}. \tag{6.1}$$

Each component is parameterized to reflect real-world deployment scenarios. The hardware manufacturing footprint C_{hw} is estimated by multiplying the embodied energy cost of server and GPU manufacturing (E_{emb}) with an emission factor EF_{manuf}, measured in kilograms of CO_2 per megawatt-hour. This enables differentiation based on hardware type, fabrication process, and sourcing location.

Infrastructure overheads are modeled using the Power Usage Effectiveness (PUE) metric. The total infrastructure-related emissions are computed as the additional energy required to support cooling, lighting, and backup systems, expressed as $(PUE - 1) \times (E_{\text{train}} + E_{\text{infer}})$, where the result is multiplied by the regional grid's carbon intensity. This term captures the compounding effect of inefficient facility operations, especially in warm or energy-constrained regions.

Training emissions, C_{train}, are computed based on the energy consumed during model training runs and the grid carbon intensity at the training site. This term varies considerably based on model size, number of training epochs, and hardware used. For example, large-scale transformer models like GPT-3 demand thousands of megawatt-hours and can produce hundreds of metric tons of CO_2 during a single training cycle. In contrast, smaller or distilled models trained on low-carbon grids have significantly reduced training footprints.

Inference emissions, C_{infer}, are sensitive to both the frequency of use and the energy per inference. We define the total inference energy as $f \cdot E_{\text{inf}} \cdot T$, where f denotes the number of inferences per day, E_{inf} is the energy per inference in kilowatt-hours, and T is the number of deployment days. This term reflects application-specific factors: real-time translation or virtual assistants generate billions of daily inferences, while specialized AI models in scientific or industrial settings may run only occasionally. The resulting inference energy is then multiplied by the local grid carbon intensity to determine its contribution to the overall emissions.

6.4.3 Parameter Analysis and Application Sensitivity

The proposed model supports analysis over a multi-dimensional parameter space. Application domain has a pronounced influence on sustainability. For instance, generative AI services such as chatbots or content recommenders exhibit high inference frequency and thus sustain a large operational footprint even if the training phase is optimized. Conversely, domains such as medical imaging or climate modeling may tolerate infrequent but high-complexity inference, shifting the footprint toward the training phase.

The carbon intensity of the deployment region plays a critical role as well. Identical models deployed in different geographies can yield vastly different emission profiles depending on grid composition. Moreover, improvements in hardware efficiency and the use of on-site renewable power or demand-shifting strategies (e.g., aligning inference loads with peak solar generation) can significantly reduce C_{infer} and C_{infra}.

6.4.4 Use Case: Emissions Estimation for a Large Language Model

To illustrate the utility of the model, consider a 175-billion-parameter language model trained with 1,300 MWh on a fossil-fuel-dominant grid with a carbon intensity of 0.4 kg CO_2 / kWh. Assuming the model is deployed for two years at one million inferences per day, with each inference consuming 0.5 Wh, the inference energy totals 365 MWh annually. With a PUE of 1.2, the infrastructure overhead adds an additional 20% to the combined training and inference energy. Plugging in these values yields a training footprint of 520,000 $kgCO_2$, an inference footprint of 292,000 $kgCO_2$, and infrastructure emissions of approximately 162,400 $kgCO_2$, resulting in a total carbon footprint exceeding 970,000 $kgCO_2$. This figure excludes hardware manufacturing, which can add tens to hundreds of additional metric tons depending on hardware scale and refresh cycles.

6.5 Analysis

Model Type	Params (B)	Train Energy (MWh)	Inference Energy (MWh)	Total Emissions (tCO_2)
Small Transformer	0.4	20	182.5	99.0
Medium Transformer	6.0	150	219.0	157.2
Large LLM	175	1300	365.0	353.6

Table 6.2: Carbon impact of model complexity (fixed inference frequency)

Setting	Model Size	Inference/Day	Total Emissions (tCO_2)
Specialist (low use, large model)	175B	10^4	547
Consumer NLP (medium use)	6B	10^6	181
Massive-scale service (high use)	0.4B	10^7	744

Table 6.3: Combined effects of model complexity and inference intensity

Scenario	Inferences/Day	Energy/Query (Wh)	Inference Emissions (tCO_2)
Mobile Assistant (low)	10^5	0.25	3.65
Chatbot (medium)	10^6	0.50	73.0
Search + Recs (high)	10^7	0.75	1,095

Table 6.4: Carbon impact of inference frequency and cost

The lifecycle emissions of an AI service are governed not only by the raw scale of compute involved, but also by high-level application characteristics that influence how models are built, deployed, and used over time. To assess these interactions, we analyze three core parameter regimes: model complexity, inference frequency, and per-query energy cost. These parameters are tightly coupled with application domains, which range from low-latency, high-traffic consumer services to infrequent, high-stakes scientific inference.

6.5.1 Effect of Model Complexity

As model complexity increases, both training energy and per-inference energy rise due to greater parameter counts and deeper architectures. Table 6.2 illustrates how increasing model size affects lifecycle emissions, assuming a fixed deployment period of one year and one million daily inferences. Carbon intensity of the grid is fixed at 0.4 kgCO_2/kWh, and a PUE of 1.2 is assumed. The results reveal that while training emissions dominate at very large model scales, inference energy still contributes substantially over time. Thus, minimizing model size—where possible—has a compound effect in reducing both upfront and operational emissions.

6.5.2 Effect of Inference Frequency and Cost per Query

In use cases such as real-time search, recommendation, or conversational AI, the number of inferences per day can reach into the tens or hundreds of millions. Table 6.4 explores how varying inference load and per-query energy cost affects overall emissions, keeping model size and training constant. The data indicates that inference emissions can rapidly become the dominant lifecycle contributor, particularly when model optimization lags behind deployment scale. Application-aware model pruning and hardware acceleration are essential in such environments.

6.5.3 Combined Sensitivity Analysis

To better understand joint parameter effects, Table 6.3 presents a scenario matrix combining model size with application-level inference patterns. The results assume a one-year deployment, 0.4 kgCO_2/kWh grid intensity, and 1.2 PUE. Interestingly, high-frequency use of small models can rival or exceed the emissions of large models used sparingly. This demonstrates that high-volume deployment can nullify the carbon savings of low training costs if inference loads are not optimized.

6.5.4 Key Takeaways

The analysis highlights several critical factors that govern the carbon footprint of AI systems across their lifecycle. First, model complexity plays a dual role, simultaneously inflating both training and inference costs. As model size scales non-linearly, training emissions can grow by an order of magnitude, yet inference remains a persistent contributor over time especially in long-lived deployments.

Inference frequency and energy per query emerge as pivotal drivers of operational emissions. Even modest per-query energy use, when coupled with high daily inference volumes, can surpass training emissions within weeks or months. Applications such as recommendation engines, chatbots, and search services exemplify this dynamic, underscoring the need for aggressive inference-side optimization.

The interplay between model size and application domain reveals important trade-offs. Small models with massive daily inference loads can generate carbon footprints comparable to large, infrequently used models. Thus, minimizing carbon emissions is not only a matter of reducing training FLOPs but also tailoring deployment strategies to usage patterns and hardware efficiency.

Finally, data center factors such as Power Usage Effectiveness (PUE) and regional grid carbon intensity remain amplifiers or suppressors of total emissions. Poor infrastructure efficiency or fossil-dominant energy sources can significantly increase lifecycle emissions, even when model and application parameters are carefully optimized. In total, the findings reinforce the necessity of lifecycle-aware, application-specific, and region-sensitive design choices in building environmentally sustainable AI systems.

6.6 Conclusion

In this chapter, we highlight how current carbon offset mechanisms such as carbon credits, Renewable Energy Certificates (RECs), and Virtual Power Purchase Agreements (VPPAs) are structurally insufficient to mitigate the rapidly escalating carbon footprint of AI data centers. These strategies fail to deliver reductions in carbon emissions at the scale and speed required, while inadvertently permitting continued reliance on non-renewable energy sources. Additionally, we design a sustainability calculator to evaluate the sustainability of a computational service and guide the system design towards minimal emissions. Without proactive intervention, AI could become one of the dominant drivers of global emissions in the coming decade, undermining international climate agreements and pushing the world closer to catastrophic environmental tipping points. It is imperative that governments, industry leaders, and regulatory bodies collaborate to establish binding requirements for renewable energy sourcing and enforce meaningful carbon accounting standards. In particular, policies should mandate that AI service deployments are contingent on verified renewable energy procurement, localized grid decarbonization efforts, and rigorous lifecycle assessments.

7 | Summary and Conclusions

The Internet has become an indispensable driver of modern life, fueling advances in education, communication, commerce, and governance. Yet, as this dissertation has shown, its benefits remain deeply unevenly distributed. While industrialized regions move rapidly toward ubiquitous 5G connectivity, immersive digital experiences, and AI-driven cloud platforms, large segments of the global population remain digitally marginalized. At the same time, the performance and sustainability of networked systems are increasingly strained by the demands of emerging applications. This dissertation addressed these dual challenges of digital inequality and network performance by designing new web access platforms and transport-layer protocols that are more inclusive, adaptive, and efficient.

7.1 Bridging the Digital Divide through Web Access Innovation

The first part of this thesis focused on understanding and mitigating the global digital divide. Our large-scale measurement study across 56 cities highlighted stark disparities in web access, where users in developing regions often pay more for slower and less reliable service while relying on low-end mobile devices. This quantitative foundation motivated the development of a suite of systems designed to make the Web lighter, more local, and more resilient.

Lite-Web demonstrated that meaningful performance improvements could be achieved on low-cost smartphones by automatically simplifying webpages in real-time. MAML provided a declarative specification language to build visually consistent and semantically meaningful web versions that render quickly on constrained hardware. These systems were deployed in GAIUS [28], an edge-based, offline-first web ecosystem that empowered community members in countries like Kenya and Bangladesh to host and access local content without needing constant connectivity. Sonic extended this work into completely offline settings by enabling users to receive web pages via FM radio and interact through SMS, with successful deployments in Cameroon.

Together, these efforts contribute to a new class of resilient, application-layer architectures tailored to the infrastructural constraints of the Global South. They underscore that addressing the digital divide requires not just more connectivity, but better content delivery systems designed for local needs, low-end devices, and intermittent networks.

7.2 Evaluating Congestion Control Protocols for Modern Networks

The second part of this dissertation turned to the transport layer, where congestion control protocols serve as a key performance lever for Internet communication. Despite advances in network infrastructure, especially the deployment of 5G, our findings show that transport protocols remain a critical bottleneck. Through the Zeus benchmarking framework, we evaluated leading congestion control algorithms (CCAs) under a range of synthetic and real 5G conditions.

This analysis revealed persistent challenges. Many widely deployed CCAs were found to underperform on key metrics such as delay, throughput stability, and fairness, particularly in dynamic cellular environments. Protocols like Cubic and BBR optimize for different trade-offs and often fail to meet the low-latency requirements of emerging interactive applications. These shortcomings highlight the need for more holistic evaluation frameworks and protocol designs that align more closely with application needs and real-world conditions.

Zeus itself contributed a new methodology for scenario-aware benchmarking, enabling direct comparison of CCAs across reproducible yet realistic network scenarios. This framework advances the ability of researchers and developers to assess transport-layer behavior under conditions representative of 5G and emerging wireless networks.

7.3 Protocol Design for Immersive and Interactive Applications

Emerging applications like virtual and augmented reality pose new challenges for transport protocols, including extreme sensitivity to delay, variability in stream priority, and high bandwidth requirements. In response, this thesis introduced Hera, an application-aware congestion control protocol co-designed with the requirements of immersive XR streaming.

Hera augments traditional rate control by incorporating feedback from the application layer, including field-of-view (FOV) prioritization and frame-level delay sensitivity. By integrating these signals, Hera adapts transmission rates in real time to optimize the user experience while maintaining network stability. Evaluation over 5G and emulated lossy links showed that Hera significantly outperforms baseline CCAs, reducing tail latency by up to 66% and improving frame quality and stability in multi-user scenarios.

This work demonstrates that co-designing transport protocols with application semantics can lead to substantial performance gains. Hera represents a step toward more intelligent, context-aware network protocols capable of supporting the next generation of interactive media and real-time collaboration tools.

7.4 Future Directions

While the primary focus of this dissertation has been on performance and accessibility, the environmental footprint of modern networked applications represents an increasingly urgent challenge. As cloud-based AI systems, immersive experiences, and edge computing proliferate, their energy demands and carbon emissions are projected to rise steeply. This trend is particularly concerning in emerging regions, where power grids are less stable, and carbon-intensive energy sources dominate.

Future work must expand the systems lens to include sustainability as a first-order design goal. This includes exploring energy-aware congestion control protocols that adjust behavior based on grid carbon intensity or time-of-day availability of renewables. Additionally, more transparent and regionally sensitive approaches to carbon accounting are needed to address the limitations of RECs and VPPAs, which often fail to deliver real or local emission reductions.

There is also a growing need to model and mitigate the indirect impacts of AI infrastructure expansion in the Global South, including energy displacement, land use for data centers, and competition with local energy needs. Aligning performance, equity, and sustainability will require collaboration across disciplines and institutions, including computer systems, policy, energy, and development sectors.

7.5 Closing Remarks

This dissertation argues that the future of the Internet must be inclusive, adaptive, and sustainable. By developing systems that improve web accessibility for the underserved and transport protocols that enable the applications of tomorrow, we take a step toward that vision. As the Internet continues to evolve, so too must the systems that support it—with a renewed focus on global equity, user experience, and environmental responsibility.

Bibliography

[1] https://www.iea.org/news/ai-is-set-to-drive-surging-electricity-demand-from-data-centres-while-offering-the-potential-to-transform-how-the-energy-sector-works. [Accessed 15-05-2025]. 2025.

[2] *#Internet4GilgitBaltistan Trends on Twitter, activists demand digital rights for Gilgit-Baltistan.* https://gbee.pk/2020/07/internet4gilgitbaltistan-trends-on-twitter-activists-demand-digital-rights-for-gilgit-baltistan/. Accessed: 2021-09-28. 2020.

[3] 3GPP. *TS 23.203: Policy and Charging Control Architecture.* 3GPP Technical Specification. .

[4] 5TONIC. *An Open Research and Innovation Laboratory Focusing on 5G Technologies.* Accessed: 19-05-2022. 2022.

[5] *A $20 phone for Africa is MWC's unlikeliest hero.* https://thenextweb.com/plugged/2019/02/26/a-20-phone-for-africa-is-mwcs-unlikeliest-hero/. Accessed: 2020-10-11. 2019.

[6] Soheil Abbasloo, Yang Xu, and H Jonathan Chao. "C2TCP: A flexible cellular TCP to meet stringent delay requirements". In: *IEEE Journal on Selected Areas in Communications* 37.4 (2019), pp. 918–932.

[7] Soheil Abbasloo, Chen-Yu Yen, and H. Jonathan Chao. "Classic Meets Modern: A Pragmatic Learning-Based Congestion Control for the Internet". In: *Proc. ACM SIGCOMM.*

Virtual Event, USA, 2020, pp. 632–647. ISBN: 9781450379557. DOI: 10.1145/3387514.3405892.

[8] Zainul Abi Din et al. “PERCIVAL: Making in-browser perceptual ad blocking practical with deep learning”. In: *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. 2020, pp. 387–400.

[9] AdBlock. *Surf the web without annoying pop ups and ads*. https://getadblock.com/. Accessed: 2020-05-02. 2009.

[10] *Affordability Report 2020*. https://1e8q3q16vyc81g8l3h3md6q5f5e-wpengine.netdna-ssl.com/wp-content/uploads/2020/12/Affordability-Report-2020.pdf. Accessed: 2021-02-14. 2020.

[11] Afrobarometer. *Africa’s shifting media landscapes: Digital media use grows, but so do demographic divides*. 2024.

[12] International Energy Agency. *Africa Energy Outlook 2022*. https://www.iea.org/reports/africa-energy-outlook-2022. 2022.

[13] Talal Ahmad et al. “Learning Congestion State For MmWave Channels”. In: *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*. mmNets’19. Los Cabos, Mexico: Association for Computing Machinery, 2019, pp. 19–25. ISBN: 9781450369329. DOI: 10.1145/3349624.3356769.

[14] Hafsa Akbar et al. *Semantic Caching for Improving Web Affordability*. 2025.

[15] Mark Allman, Vern Paxson, and Ethan Blanton. *TCP congestion control*. Tech. rep. 2009.

[16] Dario Amodei and Danny Hernandez. *AI and Compute*. OpenAI Blog. 2023.

[17] *An image format for the Web*. https://developers.google.com/speed/webp. 2023.

[18] *Android app built with Quiet which allows o pass data through the speakers of an Android device*. https://github.com/quiet/org.quietmodem.Quiet. 2023.

[19] Android Developers. *BroadcastReceiver*. Accessed: 2025-04-10. 2025.

[20] Osman Antwi-Boateng, Muhammed Danladi Musa, and Mu-Azu Iddirisu Andani and. "Audience listenership of FM radio: A case study of rural development in Northern Ghana". In: *Cogent Arts & Humanities* 10.1 (2023), p. 2184750. DOI: 10.1080/23311983.2023.2184750.

[21] Apple. *Apple News - Apple*. Web Page. Accessed: 2020-03-21. 2015.

[22] HTTP Archive. *Third Parties | 2019 | The Web Almanac by HTTP Archive*. https://almanac.httparchive.org/en/2019/third-parties. Accessed: 2020-01-2. 2019.

[23] *Arkio*. https://www.arkio.is/. Accessed: March 18, 2025. 2025.

[24] Venkat Arun and Hari Balakrishnan. "Copa: Practical delay-based congestion control for the internet". In: *Proc. of NSDI*. 2018, pp. 329–342.

[25] Evans C. Ashigwuike, Ale Felix, and Farouq E. Shaibu. "The Impact of Soil Texture on Path Loss Modelling of an FM Signal Using Diffraction Technique: A Case Study of Prime FM Radio Nigeria". In: *European Journal of Engineering Research and Science* 4.4 (Apr. 2019), pp. 56–63. DOI: 10.24018/ejers.2019.4.4.1231.

[26] Rohail Asim, Lakshmi Subramanian, and Yasir Zaki. "Impact of Congestion Control on Mixed Reality Applications". In: *Proceedings of the 2024 SIGCOMM Workshop on Emerging Multimedia Systems*. EMS '24. Sydney, NSW, Australia: Association for Computing Machinery, 2024, pp. 21–26. ISBN: 9798400707117. DOI: 10.1145/3672196.3673395.

[27] Rohail Asim et al. "Demo: Enabling High Bandwidth Applications over 5G Environments". In: *Proceedings of the ACM SIGCOMM 2024 Conference: Posters and Demos*. ACM SIGCOMM Posters and Demos '24. Sydney, NSW, Australia: Association for Computing Machinery, 2024, pp. 130–131. ISBN: 9798400707179. DOI: 10.1145/3672202.3673754.

[28] Rohail Asim et al. “The GAIUS Experience: Powering a Hyperlocal Mobile Web for Communities in Emerging Regions”. In: *Proceedings of the 13th International Conference on Information & Communication Technologies and Development*. ICTD ’24. Association for Computing Machinery, 2025, pp. 236–246. ISBN: 9798400710414. DOI: 10.1145/3700794.3700818.

[29] *AudioQR*. https://github.com/jamesonrader/AudioQR. 2023.

[30] Kalvin Bahia et al. “The Welfare Effects of Mobile Broadband Internet: Evidence from Nigeria”. In: *World Bank Policy Research Working Paper* 9230 (2020).

[31] Yang Bai et al. “BatComm: enabling inaudible acoustic communication with high-throughput for mobile devices”. In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 2020, pp. 205–217.

[32] Tina Baidar. “Signal Coverage Mapping of Local Radios”. In: *Journal on Geoinformatics, Nepal* (2017). Accessed via Nepal Journals Online.

[33] World Bank. *Purchasing Power Parities and the Real Size of World Economies*. https://data.worldbank.org/indicator/PA.NUS.PPP. 2023.

[34] Mark Bauman and Ray Bonander. *Advertisement blocker circumvention system*. US Patent App. 15/166,217. Nov. 2017.

[35] Mihai Bazon. *UglifyJS*. http://lisperator.net/uglifyjs/. Accessed: 2020-05-01. 2012.

[36] Ketan Bhardwaj et al. “DRIVESHAFT: Improving Perceived Mobile Web Performance”. In: *arXiv preprint arXiv:1809.09292* (2018).

[37] Y. Birk and D. Crupnicoff. “A multicast transmission schedule for scalable multirate distribution of bulk data using nonscalable erasure-correcting codes”. In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications*

Societies (IEEE Cat. No.03CH37428). Vol. 2. 2003, 1033–1043 vol.2. doi: 10.1109/INFCOM.2003.1208940.

[38] Radu Gabriel Bozomitu and Florin Doru Hutu. “Drivers’ Warning Application Through Personalized DSSS-CDMA Data Transmission by Using the FM Radio Broadcasting Infrastructure”. In: *IEEE Access* 11 (2023), pp. 11711–11731.

[39] Radu Gabriel Bozomitu and Ştefan Corneliu Stoica. “A Robust Radiocommunication System for FM Transmission Based on Software Defined Radio Technology”. In: *2022 IEEE 28th International Symposium for Design and Technology in Electronic Packaging (SIITME)*. IEEE. 2022, pp. 78–81.

[40] Lawrence S. Brakmo and Larry L. Peterson. “TCP Vegas: End to end congestion avoidance on a global Internet”. In: *IEEE Journal on selected Areas in communications* 13.8 (1995), pp. 1465–1480.

[41] *Brave Passes 50 Million Monthly Active Users, Growing 2x for the Fifth Year in a Row*. https://brave.com/2021-recap/. Accessed: 2022-11-11. 2022.

[42] *Brave: the Privacy Preserving Browser*. https://brave.com/. Accessed: 2022-04-20. 2020.

[43] Joe Breen et al. “POWDER: Platform for Open Wireless Data-Driven Experimental Research”. In: *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. WiNTECH’20. London, United Kingdom: Association for Computing Machinery, 2020, pp. 17–24. isbn: 9781450380829. doi: 10.1145/3411276.3412204.

[44] Tom Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[45] Michael Butkiewicz et al. “Klotski: Reprioritizing Web Content to Improve User Experience on Mobile Devices”. In: *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’15)*. Open Access. Oakland, CA, USA: USENIX Association, 2015. ISBN: 978-1-931971-218.

[46] Neal Cardwell et al. “BBR: Congestion-Based Congestion Control”. In: *Queue* 14.5 (Oct. 2016), 50:20–50:53. ISSN: 1542-7730. DOI: 10.1145/3012426.3022184.

[47] Neal Cardwell et al. “Bbrv2: A model-based congestion control”. In: *Presentation in ICCRG at IETF 104th meeting*. 2019.

[48] Gaetano Carlucci, Luca De Cicco, and Saverio Mascolo. “Controlling queuing delays for real-time communication: the interplay of E2E and AQM algorithms”. In: *SIGCOMM Comput. Commun. Rev.* 46.3 (July 2018). ISSN: 0146-4833. DOI: 10.1145/3243157.3243158.

[49] Gaetano Carlucci et al. “Analysis and design of the google congestion control for web real-time communication (WebRTC)”. In: *Proceedings of the 7th International Conference on Multimedia Systems*. MMSys ’16. Klagenfurt, Austria: Association for Computing Machinery, 2016. ISBN: 9781450342971. DOI: 10.1145/2910017.2910605.

[50] Giovanna Carofiglio et al. “A hands-on assessment of transport protocols with lower than best effort priority”. In: *IEEE Local Computer Network Conference*. IEEE. 2010, pp. 8–15.

[51] Miguel Casasnovas et al. “Experimental evaluation of interactive edge/cloud virtual reality gaming over wi-fi using unity render streaming”. In: *Computer Communications* 226 (2024), p. 107919.

[52] Central Statistics Office Ireland. *Metered Electricity Consumption 2023*. Accessed: 2025-07-10. Government of Ireland, 2024.

[53] Xintao Chai et al. “Deep learning for irregularly and regularly missing data reconstruction”. In: *Scientific Reports* 10.1 (Feb. 2020), p. 3302. ISSN: 2045-2322. DOI: 10.1038/s41598-020-59801-x.

[54] Moumena Chaqfeh et al. “Jsanalyzer: A web developer tool for simplifying mobile web pages through non-critical javascript elimination”. In: *ACM Transactions on the Web* 16.4 (2022), pp. 1–31.

[55] Moumena Chaqfeh et al. “JSCleaner: De-Cluttering Mobile Webpages Through JavaScript Cleanup”. In: *Proceedings of The Web Conference 2020*. 2020, pp. 763–773.

[56] Moumena Chaqfeh et al. “To Block or Not to Block: Accelerating Mobile Web Pages On-The-Fly Through JavaScript Classification”. In: *arXiv preprint arXiv:2106.13764* (2021).

[57] Moumena Chaqfeh et al. “To Block or Not to Block: Accelerating Mobile Web Pages On-The-Fly Through JavaScript Classification”. In: *CoRR* abs/2106.13764 (2021).

[58] Moumena Chaqfeh et al. “To Block or Not to Block: Accelerating Mobile Web Pages On-The-Fly Through JavaScript Classification”. In: *Proceedings of the 12th International Conference on Information and Communication Technologies and Development, ICTD 2022, Seattle, USA, June 27-29, 2022 (accepted)*. 2022.

[59] Moumena Chaqfeh et al. “Towards a World Wide Web without digital inequality”. In: *Proceedings of the National Academy of Sciences* 120.3 (2023), e2212649120. DOI: 10.1073/pnas.2212649120.

[60] Joung-min Cho et al. “78-4: Screen door effect mitigation and its quantitative evaluation in VR display”. In: *SID symposium digest of technical papers*. Vol. 48. 1. Wiley Online Library. 2017, pp. 1154–1156.

[61] Ravi Chugh et al. “Staged Information Flow for Javascript”. In: *SIGPLAN Not.* 44.6 (June 2009), pp. 50–62. ISSN: 0362-1340. DOI: 10.1145/1543135.1542483.

[62] CircleCell. *JSCompress - The JavaScript Compression Tool*. https://jscompress.com/. Accessed: 2020-05-01. 2011.

[63] *CITI Program - Collaborative Institutional Training Initiative*. www.citiprogram.org. Accessed: 2022-05-18. 2019.

[64] Google Cloud. *Google Cloud Pricing Calculator*. https://cloud.google.com/products/calculator. 2024.

[65] Cloudflare, Inc. *Threat Intelligence APIs - Security Center*. Accessed: 2025-03-22. 2025.

[66] *Connecting Africa Through Broadband A strategy for doubling connectivity by 2021 and reaching universal access by 2030*. https://www.broadbandcommission.org/Documents/working-groups/DigitalMoonshotforAfrica_Report.pdf. Accessed: 2020-10-21. 2019.

[67] Corbet. “Pluggable congestion avoidance modules”. In: *LWN.net: https://lwn.net/Articles/128681/* (2005).

[68] Roberto Irajá Tavares da Costa Filho et al. “Predicting the performance of virtual reality video streaming in mobile networks”. In: *Proceedings of the 9th ACM Multimedia Systems Conference*. MMSys ’18. Amsterdam, Netherlands: Association for Computing Machinery, 2018, pp. 270–283. ISBN: 9781450351928. DOI: 10.1145/3204949.3204966.

[69] Counterpoint Research. “Global Smartphone Market Share: Quarterly”. In: (2025).

[70] Paul Covington, Jay Adams, and Emre Sargin. “Deep Neural Networks for YouTube Recommendations”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys ’16. Boston, Massachusetts, USA: Association for Computing Machinery, 2016, pp. 191–198. ISBN: 9781450340359. DOI: 10.1145/2959100.2959190.

[71] Josh Cowls et al. “The AI Gambit: Leveraging Artificial Intelligence to Combat Climate Change”. In: *Nature Machine Intelligence* 3 (2021), pp. 589–599.

[72] Sybu Data. *Sybu JavaScript Blocker – Google Chrome Extension*. https://sybu.co.za/wp/projects/js-blocker/. Accessed: 2020-05-02. 2016.

[73] DataReportal. *Digital 2025: Cameroon*. Accessed: 2025-03-26. 2025.

[74] Alex de Vries. "The growing energy footprint of artificial intelligence". In: *Joule* 7.10 (2023), pp. 2191–2194. ISSN: 2542-4351. DOI: https://doi.org/10.1016/j.joule.2023.09.004.

[75] Laura Derksen, Catherine Michaud Leclerc, Pedro CL Souza, et al. *Searching for answers: The impact of student access to wikipedia*. University of Warwick, Centre for Competitive Advantage in the Global ..., 2019.

[76] Luis Diez et al. "Can We Exploit Machine Learning to Predict Congestion over mmWave 5G Channels?" In: *Applied Sciences* 10.18 (2020), p. 6164.

[77] Luis Diez et al. "Learning congestion over millimeter-wave channels". In: *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2020, pp. 1–6. DOI: 10.1109/WiMob50308.2020.9253443.

[78] Inc. Docker. *Docker: Open platform for developing, shipping, and running applications*. 2024.

[79] Jesse Dodge et al. "Measuring the Carbon Intensity of AI in Cloud Instances". In: *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2022, pp. 1877–1894.

[80] Mo Dong et al. "PCC vivace: Online-learning congestion control". In: *Proc. of NSDI*. 2018, pp. 343–356.

[81] Mo Dong et al. "PCC: Re-architecting congestion control for consistent high performance". In: *Proc. of NSDI*. 2015, pp. 395–408.

[82] Eric Elliott. *How Popular is JavaScript in 2019?* https://medium.com/javascript-scene/how-popular-is-javascript-in-2019-823712f7c4b1. Accessed: 2020-11-08. 2019.

[83] European Commission. *Carbon Border Adjustment Mechanism*. European Union Policy. 2023.

[84] *Expanding internet connectivity with stratospheric balloons.* https://x.company/projects/loon/. 2023.

[85] Facebook. *Instant Articles | Facebook.* Web Page. Accessed: 2020-03-21. 2015.

[86] Roberto Iraja Tavares Da Costa Filho et al. "Dissecting the Performance of VR Video Streaming through the VR-EXP Experimentation Platform". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 15.4 (Dec. 2019). ISSN: 1551-6857. DOI: 10.1145/3360286.

[87] Fraunhofer FOKUS. *5G Playground.* Accessed: 19-05-2022. 2022.

[88] Jim Gettys and Kathleen Nichols. "Bufferbloat: dark buffers in the internet". In: *Communications of the ACM* 55.1 (2012), pp. 57–65.

[89] *GGwave.* https://github.com/ggerganov/ggwave. 2023.

[90] Mohammad Ghasemisharif et al. "SpeedReader: Reader Mode Made Fast and Private". In: *CoRR* abs/1811.03661 (2018).

[91] Mohammad Ghasemisharif et al. "Speedreader: Reader mode made fast and private". In: *The World Wide Web Conference.* 2019, pp. 526–537.

[92] Google. *24/7 Carbon-Free Energy: Methodology and Metrics.* Tech. rep. Google LLC, 2022.

[93] Google. *AMP is a web component framework to easily create user-first web experiences - amp.dev.* https://amp.dev. Accessed: 2019-05-05. 2019.

[94] Prateesh Goyal et al. "ABC: A Simple Explicit Congestion Controller for Wireless Networks". In: *Proc. of NSDI).* Santa Clara, CA, Feb. 2020, pp. 353–372. ISBN: 978-1-939133-13-7.

[95] Aaron Grattafiori, Laurens Van Der Maaten, et al. *The LLaMA 3 Herd of Models.* https://arxiv.org/abs/2407.21783. 2024.

[96] GSMA. *The State of Mobile Internet Connectivity Report 2024.* Accessed: 2025-02-05. 2024.

[97] *GSMA Connected Society. The State of Mobile Internet Connectivity 2019.* https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2019/07/GSMA-State-of-Mobile-Internet-Connectivity-Report-2019.pdf. Accessed: 2020-10-04. 2019.

[98] GSMArena. *GSMArena - Mobile Phone Specifications, News, Reviews.* Accessed: 21 Mar. 2025. 2025.

[99] Tian Guo. "Resource-Efficient and Privacy-Preserving Edge for Augmented Reality". In: *Proceedings of the 2023 Workshop on Emerging Multimedia Systems.* EMS '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 22–27. ISBN: 9798400703034. DOI: 10.1145/3609395.3610596.

[100] Sangtae Ha, Injong Rhee, and Lisong Xu. "CUBIC: A New TCP-Friendly High-Speed TCP Variant". In: *SIGOPS Oper. Syst. Rev.* 42.5 (July 2008), pp. 64–74. ISSN: 0163-5980. DOI: 10.1145/1400097.1400105.

[101] Rumaisa Habib et al. "A Framework for Improving Web Affordability and Inclusiveness". In: *Proceedings of the ACM SIGCOMM 2023 Conference.* ACM SIGCOMM '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 592–607. ISBN: 9798400702365. DOI: 10.1145/3603269.3604872.

[102] Habtegebreil Haile et al. "End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks". In: *Computer Networks* 186 (2021), p. 107692.

[103] Habtegebreil Haile et al. "Performance of QUIC congestion control algorithms in 5G networks". In: *Proceedings of the ACM SIGCOMM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases.* 2022, pp. 15–21.

[104] Nikhil Handigol et al. "Reproducible network experiments using container-based emulation". In: *Proceedings of the 8th international conference on Emerging networking experiments and technologies.* 2012, pp. 253–264.

[105] John D Hansen and Justin Reich. “Democratizing education? Examining access and usage patterns in massive open online courses”. In: *Science* 350.6265 (2015), pp. 1245–1248.

[106] Waleed Hashmi et al. “PQual: Automating Web Pages Qualitative Evaluation”. In: UIST ’20 Adjunct. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 148–150. ISBN: 9781450375153. DOI: 10.1145/3379350.3416163.

[107] Simon Hearne. *Third Parties | 2020 | The Web Almanac by HTTP Archive*. https://almanac.httparchive.org/en/2020/third-parties. Accessed: 2021-09-26. 2020.

[108] S Hemminger. “TCP infrastructure split out”. In: *Mailing list: http://goo. gl/xYYWml (gmane. org)* 34 (2005).

[109] Stephen Hemminger et al. “Network emulation with NetEm”. In: *Linux conf au*. Vol. 844. Citeseer. 2005.

[110] Peter Henderson et al. “Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning”. In: *Journal of Machine Learning Research* 24.105 (2023), pp. 1–43.

[111] *High altitude connectivity: The next chapter*. https://engineering.fb.com/2018/06/27/connectivity/high-altitude-connectivity-the-next-chapter/. 2023.

[112] Jonas Hjort and Jonas Poulsen. “The arrival of fast internet and employment in Africa”. In: *American Economic Review* 109.3 (2019), pp. 1032–79.

[113] Xueshi Hou et al. “5G and AR/VR bring local kids center court at Kings game”. In: (2018).

[114] Huawei Technologies Co., Ltd. *Mobile WIFI 4G Router Huawei E8372h-517 WiFi Modem*. Online. Accessed: 25 March 2025.

[115] Apple Inc. “Apple Vision Pro Available in the US on February 2”. In: (2024). Accessed: 2025-03-17.

[116] Apple Inc. “Apple Vision Pro Brings a New Era of Spatial Computing to Business”. In: (2024). Accessed: 2025-03-17.

[117] Apple Inc. “Apple Vision Pro Unlocks New Opportunities for Health App Developers”. In: (2024). Accessed: 2025-03-17.

[118] Meta Inc. “Accelerating the Future: AI, Mixed Reality, and the Metaverse”. In: (2024). Accessed: 2025-03-17.

[119] *Information and Communication Technologies for Women’s Socioeconomic Empowerment.* http://documents1.worldbank.org/curated/fr/812551468148179172/pdf/518310PUB0REPL101Official0Use0Only1.pdf. Accessed: 2021-02-4. 2009.

[120] GSMA Intelligence. *The Mobile Economy: Sub-Saharan Africa 2023.* https://www.gsma.com/mobileeconomy/sub-saharan-africa/. 2023.

[121] Cliqz International. *Ghostery Makes the Web Cleaner, Faster and Safer.* https://www.ghostery.com/. Accessed: 2020-05-2. 2009.

[122] International Energy Agency. *Data Centres and Energy Demand.* Tech. rep. IEA, 2023.

[123] *Internet Archive Wayback Machine.* https://web.archive.org/. Accessed: 2020-07-21. 2014.

[124] *Introducing the world’s most affordable smart feature phone – The Digit 4G.* https://www.kaiostech.com/introducing-the-worlds-most-affordable-smart-feature-phone-the-digit-4g/. Accessed: 2020-10-11. 2020.

[125] Teerawat Issariyakul and Ekram Hossain. “Introduction to network simulator 2 (NS2)”. In: *Introduction to network simulator NS2.* Springer, 2009, pp. 1–18.

[126] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. “A quantitative measure of fairness and discrimination”. In: *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA* 21 (1984).

[127] Nathan Jay et al. "A deep reinforcement learning perspective on internet congestion control". In: *International Conference on Machine Learning*. 2019, pp. 3050–3059.

[128] Robert Jensen. "The Digital Provide: Information (Technology), Market Performance, and Welfare in the South Indian Fisheries Sector". In: *The Quarterly Journal of Economics* 122.3 (2007), pp. 879–924. ISSN: 00335533, 15314650.

[129] Haiqing Jiang et al. "Understanding bufferbloat in cellular networks". In: *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*. 2012, pp. 1–6.

[130] *Jio likely to launch affordable Android phones in India by Dec 2020: Report*. https://www.bgr.in/news/jio-launch-india-android-phone-low-cost-2020-price-more-913657/. Accessed: 2020-10-11. 2020.

[131] Ingemar Johansson. "Self-clocked rate adaptation for conversational video in LTE". In: *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*. CSWS '14. Chicago, Illinois, USA: Association for Computing Machinery, 2014, pp. 51–56. ISBN: 9781450329910. DOI: 10.1145/2630088.2631976.

[132] Soonwon Ka et al. "Near-ultrasound communication for tv's 2nd screen services". In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 2016, pp. 42–54.

[133] Tomoaki Kanaya, Nobuo Tabata, and Saneyasu Yamaguchi. "A study on performance of CUBIC TCP and TCP BBR in 5G environment". In: *2020 IEEE 3rd 5G World Forum (5GWF)*. IEEE. 2020, pp. 508–513.

[134] Conor Kelton et al. "Browselite: A Private Data Saving Solution for the Web". In: *arXiv preprint arXiv:2102.07864* (2021).

[135] *Keyword Research, Competitor Analysis, and Website Ranking | Alexa*. https://www.alexa.com/. Accessed: 2019-11-04. 1996-2020.

[136] Muhammad Khan et al. "The case for model-driven interpretability of delay-based congestion control protocols". In: *ACM SIGCOMM Computer Communication Review* 51.1 (2021), pp. 18–25.

[137] Matt Kimball. *My traceroute (MTR)*. https://www.bitwizard.nl/mtr/.

[138] K. Kondepu et al. "Experimental Demonstration of 5G Virtual EPC Recovery in Federated Testbeds". In: *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. 2019, pp. 712–713.

[139] Paul R Krugman and Maurice Obstfeld. *International economics: Theory and policy*. Pearson Education, 2009.

[140] Umakant Kulkarni et al. "Understanding the Impact of Wi-Fi Configuration on Volumetric Video Streaming Applications". In: *Proceedings of the 2023 Workshop on Emerging Multimedia Systems*. EMS '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 41–47. ISBN: 9798400703034. DOI: 10.1145/3609395.3610599.

[141] Jesutofunmi Kupoluyi et al. "Muzeel: assessing the impact of JavaScript dead code elimination on mobile web performance". In: *Proceedings of the 22nd ACM Internet Measurement Conference*. 2022, pp. 335–348.

[142] Tofunmi Kupoluyi et al. "Muzeel: A Dynamic JavaScript Analyzer for Dead Code Elimination in Today's Web". In: *arXiv preprint arXiv:2106.08948* (2021).

[143] A. Kuzmanovic and E.W. Knightly. "TCP-LP: a distributed algorithm for low priority data transfer". In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*. Vol. 3. 2003, 1691–1701 vol.3. DOI: 10.1109/INFCOM.2003.1209192.

[144] Richard J La, Jean Walrand, and Venkatachalam Anantharam. *Issues in TCP vegas*. Citeseer, 1999.

[145] Adam Langley et al. “The QUIC Transport Protocol: Design and Internet-Scale Deployment”. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM ’17. Los Angeles, CA, USA: Association for Computing Machinery, 2017, pp. 183–196. ISBN: 9781450346535. DOI: 10.1145/3098822.3098842.

[146] Victor Le Pochat et al. “Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation”. In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. NDSS 2019. Feb. 2019. DOI: 10.14722/ndss.2019.23386.

[147] Hyewon Lee et al. “Chirp signal-based aerial acoustic communication for smart devices”. In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2015, pp. 2407–2415.

[148] Chih-Ping Li et al. “5G ultra-reliable and low-latency systems design”. In: *2017 European Conference on Networks and Communications (EuCNC)*. IEEE. 2017, pp. 1–5.

[149] Lili Li et al. “E-commerce development and urban-rural income gap: Evidence from Zhejiang Province, China”. In: *Papers in Regional Science* 100.2 (2021), pp. 475–494.

[150] *Lighthouse*. https://developer.chrome.com/docs/lighthouse.

[151] LineageOS. *LineageOS Android Distribution*. A free and open-source operating system for various devices, based on the Android mobile platform. 2024.

[152] Xing Liu et al. “Firefly: Untethered Multi-user VR for Commodity Mobile Devices”. In: *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. USENIX Association, July 2020, pp. 943–957. ISBN: 978-1-939133-14-4.

[153] *Load mobile pages faster with Web Light*. https://support.google.com/websearch/answer/9836344?hl=en. 2023.

[154] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model*. 2022.

[155] Yiqing Ma et al. “Multi-Objective Congestion Control”. In: *Proc. 17th EuroSys Conf.* Rennes, France: Association for Computing Machinery, 2022, pp. 218–235. ISBN: 9781450391627. DOI: 10.1145/3492321.3519593.

[156] Jonathan Mace et al. “Retro: Targeted resource management in multi-tenant distributed systems”. In: *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*. NSDI’15. Oakland, CA: USENIX Association, 2015, pp. 589–603. ISBN: 9781931971218.

[157] *Making Mobile Internet Technology More Affordable*. https://www.newpath.com/. Accessed: 2020-10-11. 2020.

[158] Ivano Malavolta et al. “JavaScript Dead Code Identification, Elimination, and Empirical Assessment”. In: *IEEE Transactions on Software Engineering* (2023).

[159] Simone Mangiante et al. “Congestion Control for Future Mobile Networks”. In: *Proceedings of the 13th Workshop on Challenged Networks*. 2018, pp. 55–61.

[160] Simone Mangiante et al. “VR is on the Edge: How to Deliver 360° Videos in Mobile Networks”. In: Aug. 2017, pp. 30–35. DOI: 10.1145/3097895.3097901.

[161] Vuk Marojevic et al. “Advanced Wireless for Unmanned Aerial Systems: 5G Standardization, Research Challenges, and AERPAW Experimentation Platform”. In: *IEEE Vehicular Technology Magazine* PP (Apr. 2020). DOI: 10.1109/MVT.2020.2979494.

[162] Ferran Maura, Miguel Casasnovas, and Boris Bellalta. “Experimenting with Adaptive Bitrate Algorithms for Virtual Reality Streaming over Wi-Fi”. In: *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. ACM MobiCom ’24. Washington D.C., DC, USA: Association for Computing Machinery, 2024, pp. 1930–1937. ISBN: 9798400704895. DOI: 10.1145/3636534.3697322.

[163] Bryan McQuade and Barry Pollard. *How the Core Web Vitals metrics thresholds were defined*. Accessed: 2025-04-16. 2020.

[164] Tong Meng et al. "PCC proteus: Scavenger transport and beyond". In: *Proc. ACM SIGCOMM*. 2020, pp. 615–631.

[165] *Meta Horizon Worlds*. Accessed on Mar 17, 2025. 2025.

[166] Marco Mezzavilla et al. "End-to-End Simulation of 5G mmWave Networks". In: *IEEE Communications Surveys Tutorials* 20.3 (2018), pp. 2237–2263. DOI: 10.1109/COMST.2018.2828880.

[167] Radhika Mittal et al. "TIMELY: RTT-based congestion control for the datacenter". In: *ACM SIGCOMM Computer Communication Review* 45.4 (2015), pp. 537–550.

[168] *More Africans have access to cell phone service than piped water*. https://edition.cnn.com/2016/01/19/africa/africa-afrobarometer-infrastructure-report/index.html. Accessed: 2020-10-11. 2016.

[169] Sebastian Moss. "Microsoft signs 900 MW PPAs for Ireland, 28% of nation's target for 2030". In: *Data Center Dynamics* (2022). Accessed via DataCenterDynamics search results.

[170] *MSN Direct*. https://en.wikipedia.org/wiki/MSN_Direct. 2023.

[171] MTN Cameroon. *MTN Cameroon Official Website*. Accessed: 2025-03-20. 2025.

[172] Usama Naseer, Theophilus A Benson, and Ravi Netravali. "WebMedic: Disentangling the Memory-Functionality Tension for the Next Billion Mobile Web Users". In: *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications*. 2021, pp. 71–77.

[173] R. Netravali et al. "Mahimahi: Accurate Record-and-Replay for HTTP." In: *USENIX Annual Technical Conference*. 2015, pp. 417–429.

[174] Ravi Netravali and James Mickens. "Prophecy: Accelerating Mobile Page Loads Using Final-state Write Logs". In: *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. Renton, WA: USENIX Association, Apr. 2018, pp. 249–266. ISBN: 978-1-939133-01-4.

[175] Ravi Netravali et al. "Mahimahi: A Lightweight Toolkit for Reproducible Web Measurement". In: *ACM SIGCOMM* 44.4 (Aug. 2014), pp. 129–130. ISSN: 0146-4833. DOI: 10.1145/2740070.2631455.

[176] Ravi Netravali et al. "Polaris: Faster Page Loads Using Fine-grained Dependency Tracking". In: *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. Santa Clara, CA: USENIX Association, 2016.

[177] Yong Niu et al. "A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges". In: *Wireless networks* 21.8 (2015), pp. 2657–2676.

[178] James O'Connor. *Renewable energy at the heart of Microsoft's sustainability journey.* https://pulse.microsoft.com/en-ie/sustainable-futures-en-ie/na/fa1-renewable-energy-at-the-heart-of-microsofts-sustainability-journey/. Accessed 2025-07-10. 2022.

[179] Niels Groot Obbink et al. "An extensible approach for taming the challenges of JavaScript dead code elimination". In: *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE. 2018, pp. 291–401.

[180] OOKLA. *Speedtest® CLI, Internet connection measurement for developers*. https://www.speedtest.net/apps/cli.

[181] OpenAI. *OpenAI API Reference*. Accessed: 2025-03-24. 2024.

[182] *Opera Mini*. https://www.opera.com/browsers/opera-mini. Accessed: 2022-04-20. 2006.

[183] *Opera Mini for Android*. https://www.opera.com/mobile/mini. Accessed: 2021-01-11.

[184] Addy Osmani. *The cost of JavaScript*. https://medium.com/@addyosmani/the-cost-of-javascript-in-2018-7d8950fbb5d4. Accessed: 2019-05-05. 2018.

[185] Ayush Pandey et al. "Demo: Towards Faster Web in Developing Regions". In: *Proceedings of the ACM SIGCOMM 2024 Conference: Posters and Demos*. ACM SIGCOMM Posters and Demos '24. Sydney, NSW, Australia: Association for Computing Machinery, 2024, pp. 127–129. ISBN: 9798400707179. DOI: 10.1145/3672202.3673751.

[186] Ayush Pandey et al. "MAML: Towards a Faster Web in Developing Regions". In: *Proceedings of the ACM on Web Conference 2025*. WWW '25. Sydney NSW, Australia: Association for Computing Machinery, 2025, pp. 727–739. ISBN: 9798400712746. DOI: 10.1145/3696410.3714584.

[187] Ayush Pandey et al. "SONIC: Connect the Unconnected via FM Radio & SMS". In: *Proceedings of the 20th International Conference on Emerging Networking EXperiments and Technologies*. CoNEXT '24. Los Angeles, CA, USA: Association for Computing Machinery, 2024, pp. 41–47. ISBN: 9798400711084. DOI: 10.1145/3680121.3697812.

[188] Ayush Pandey et al. "SONIC: Connect the Unconnected via FM Radio & SMS". In: *Proceedings of the 20th International Conference on Emerging Networking EXperiments and Technologies*. CoNEXT '24. Los Angeles, CA, USA: Association for Computing Machinery, 2024, pp. 41–47. ISBN: 9798400711084. DOI: 10.1145/3680121.3697812.

[189] David Patterson et al. "Carbon Emissions and Large Neural Network Training". In: *arXiv preprint arXiv:2204.05149* (2022).

[190] Cristina Perfecto et al. "Taming the Latency in Multi-User VR 360°: A QoE-Aware Deep Learning-Aided Multicast Framework". In: *IEEE Transactions on Communications* 68.4 (Apr. 2020), pp. 2491–2508. ISSN: 1558-0857. DOI: 10.1109/tcomm.2020.2965527.

[191] Andrea Pinto et al. “Hercules: An Emulation-Based Framework for Transport Layer Measurements over 5G Wireless Networks”. In: *Proceedings of the 17th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization*. 2023, pp. 72–79.

[192] *Prolific | Quickly find research participants you can trust*. `https://www.prolific.com/`. Accessed: 2025-04-12.

[193] Zhen Qin et al. “Image inpainting based on deep learning: A review”. In: *Displays* 69 (2021), p. 102028. ISSN: 0141-9382. DOI: `https://doi.org/10.1016/j.displa.2021.102028`.

[194] *Quiet modem project*. `https://github.com/quiet/quiet`. 2023.

[195] Jack W. Rae et al. “Scaling Laws for Carbon Emissions”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022, pp. 23405–23418.

[196] T. S. Rappaport et al. “Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!” In: *IEEE Access* 1 (2013), pp. 335–349.

[197] Rayhan Rashed et al. “Bridging the Last Mile: Unpacking the Rural Digital Divide in Bangladesh”. In: *Proceedings of the 2025 ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*. COMPASS ’25. Association for Computing Machinery, 2025, pp. 150–166. ISBN: 9798400714849. DOI: `10.1145/3715335.3735463`.

[198] Dipankar Raychaudhuri et al. “Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless”. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–13.

[199] *Rec Room - Play, Create, and Connect with Friends*. `https://recroom.com/`. Accessed: March 18, 2025. 2025.

[200] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: `10.1109/CVPR.2016.91`.

[201] *Report: State of JavaScript.* https://httparchive.org/reports/state-of-javascript. Accessed: 2020-11-08. 2020.

[202] *Report: State of the Web.* https://httparchive.org/reports/state-of-the-web. Accessed: 2020-11-08. 2020.

[203] World Population Review. *Cost of Electricity by Country.* https://worldpopulationreview.com/country-rankings/cost-of-electricity-by-country. 2024.

[204] Rick and Morty Wiki Contributors. *Dimension C-137.* Accessed: 24 Mar. 2025. 2025.

[205] George F. Riley and Thomas R. Henderson. "The ns-3 Network Simulator." In: *Modeling and Tools for Network Simulation.* Ed. by Klaus Wehrle, Mesut Günes, and James Gross. Springer, 2010, pp. 15–34. ISBN: 978-3-642-12330-6.

[206] William J Ripple et al. "The 2024 state of the climate report: Perilous times on planet Earth". In: *BioScience* 74.12 (Oct. 2024), pp. 812–824. ISSN: 1525-3244. DOI: 10.1093/biosci/biae087.

[207] Luigi Rizzo. "Dummynet: a simple approach to the evaluation of network protocols". In: *ACM SIGCOMM Computer Communication Review* 27.1 (1997), pp. 31–41.

[208] Travis Roman. *JS Blocker.* https://jsblocker.toggleable.com/. Accessed: 2020-05-02. 2018.

[209] Dario Rossi et al. "LEDBAT: the new BitTorrent congestion control protocol". In: *2010 Proceedings of 19th International Conference on Computer Communications and Networks.* IEEE. 2010, pp. 1–6.

[210] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. "Backdoor: Making microphones hear inaudible sounds". In: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services.* 2017, pp. 2–14.

[211] Olivier Rukundo and Hanqiang Cao. “Nearest Neighbor Value Interpolation”. In: *International Journal of Advanced Computer Science and Applications* 3.4 (2012). DOI: 10.14569/IJACSA.2012.030405.

[212] Kimberly Ruth et al. “A world wide view of browsing the world wide web”. In: *Proceedings of the 22nd ACM Internet Measurement Conference*. IMC ’22. Nice, France: Association for Computing Machinery, 2022, pp. 317–336. ISBN: 9781450392594. DOI: 10.1145/3517745.3561418.

[213] *Safari Reader Mode*. https://support.apple.com/guide/safari/hide-ads-when-reading-sfri32632/mac. 2023.

[214] Chiranjib Saha and Harpreet S Dhillon. “Millimeter wave integrated access and backhaul in 5G: Performance analysis and design insights”. In: *IEEE Journal on Selected Areas in Communications* 37.12 (2019), pp. 2669–2684.

[215] Salamek. *Huawei LTE API*. GitHub repository. Accessed: 25 March 2025. 2025.

[216] Constantin Sander et al. “DeePCCI: Deep Learning-Based Passive Congestion Control Identification”. In: *Proceedings of the 2019 Workshop on Network Meets AI & ML*. NetAI’19. Beijing, China: Association for Computing Machinery, 2019, pp. 37–43. ISBN: 9781450368728. DOI: 10.1145/3341216.3342211.

[217] G Enrico Santagati and Tommaso Melodia. “A software-defined ultrasonic networking framework for wearable devices”. In: *IEEE/ACM Transactions on Networking* 25.2 (2016), pp. 960–973.

[218] Aaron Schulman, Dave Levin, and Neil Spring. “RevCast: Fast, private certificate revocation over FM radio”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 799–810.

[219] Roy Schwartz et al. “Green AI”. In: *Communications of the ACM* 63.12 (2020), pp. 54–63.

[220] *SeleniumHQ Browser Automation*. 2017. URL: http://www.seleniumhq.org/about/ (visited on 09/17/2017).

[221] Sea Shalunov et al. *Low extra delay background transport (LEDBAT)*. Tech. rep. 2012.

[222] *Six new partners to deliver affordable smart feature phones running on KaiOS in Africa*. https://www.kaiostech.com/press/six-new-partners-to-deliver-affordable-smart-feature-phones-running-on-kaios-in-africa/. Accessed: 2020-10-11. 2020.

[223] Christopher Slezak et al. "Understanding end-to-end effects of channel dynamics in millimeter wave 5G new radio". In: *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE. 2018, pp. 1–5.

[224] We Are Social and Meltwater. *Digital 2025 Global Overview Report*. Retrieved on 18 March 2025. 2025.

[225] *Spatial: Collaborate in 3D*. https://www.spatial.io/. Accessed: March 18, 2025. 2025.

[226] Isidora Stanković. "Recovery of Images with Missing Pixels using a Gradient Compressive Sensing Algorithm". In: *ArXiv* abs/1407.3695 (2014).

[227] *Starlink*. https://starlink.com/. 2023.

[228] Schmida Steve et al. *CONNECTING THE NEXT FOUR BILLION: STRENGTHENING THE GLOBAL RESPONSE FOR UNIVERSAL INTERNET ACCESS*. https://www.usaid.gov/sites/default/files/documents/15396/Connecting_the_Next_Four_Billion.pdf. Accessed: 2020-10-11. 2017.

[229] Jakob Struye et al. "Opportunities and Challenges for Virtual Reality Streaming over Millimeter-Wave: An Experimental Analysis". In: *2022 13th International Conference on Network of the Future (NoF)*. IEEE, Oct. 2022, pp. 1–5. DOI: 10.1109/nof55974.2022.9942535.

[230] Jakob Struye et al. “Toward Interactive Multi-User Extended Reality Using Millimeter-Wave Networking”. In: *IEEE Communications Magazine* 62.8 (Aug. 2024), pp. 54–60. ISSN: 1558-1896. DOI: 10.1109/mcom.001.2300804.

[231] *Students in Gilgit Baltistan protest as they suffer due to poor Internet quality.* https://newsvibesofindia.com/students-gilgit-baltistan-protest-suffer-due-to-poor-internet-quality-28006/. Accessed: 2021-09-28. 2020.

[232] Kyoungwon Suh et al. “Push-to-peer video-on-demand system: Design and evaluation”. In: *IEEE Journal on Selected Areas in Communications* 25.9 (2007), pp. 1706–1716.

[233] Shu Sun, George R. MacCartney, and Theodore S. Rappaport. “A novel millimeter-wave channel simulator and applications for 5G wireless communications”. In: *2017 IEEE Int. Conf. Commun. (ICC)*. 2017, pp. 1–7. DOI: 10.1109/ICC.2017.7996792.

[234] *Taara: A Google X Moonshot.* Accessed: 2025-03-19. 2025.

[235] Yu Wei Tan et al. “DHR+S: distributed hybrid rendering with realistic real-time shadows for interactive thin client metaverse and game applications”. In: *Vis. Comput.* 40.7 (June 2024), pp. 4981–4991. ISSN: 0178-2789. DOI: 10.1007/s00371-024-03501-4.

[236] *The age of digital interdependence.* https://digitalcooperation.org/wp-content/uploads/2019/06/DigitalCooperation-report-web-FINAL-1.pdf. Accessed: 2020-10-04. 2019.

[237] The White House. *Inflation Reduction Act.* U.S. Government Policy. 2022.

[238] Financial Times. “Meta’s Investment in VR and Smart Glasses on Track to Top $100bn”. In: (2024). Accessed: 2025-03-17.

[239] Rahul Dev Tripathi, Minzhao Lyu, and Vijay Sivaraman. “ Assessing the Impact of Network Quality-of-Service on Metaverse Virtual Reality User Experience ”. In: *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom).*

Los Alamitos, CA, USA: IEEE Computer Society, Aug. 2024, pp. 206–213. DOI: 10.1109/MetaCom62920.2024.00042.

[240] U.S. Energy Information Administration. *Electric Power Monthly*. 2023.

[241] UNESCO. *Recommendation on the Ethics of Artificial Intelligence*. https://unesdoc.unesco.org/ark:/48223/pf0000381137. 2021.

[242] Andras Varga. "OMNeT++". In: *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

[243] Matteo Varvello, Hyunseok Chang, and Yasir Zaki. "Performance characterization of videoconferencing in the wild". In: *Proceedings of the 22nd ACM Internet Measurement Conference*. 2022, pp. 261–273.

[244] Matteo Varvello and Yasir Zaki. "A Worldwide Look Into Mobile Access Networks Through the Eyes of AmiGos". In: *TMA 2023 - Proceedings of the 7th Network Traffic Measurement and Analysis Conference*. Institute of Electrical and Electronics Engineers Inc. DOI: 10.23919/TMA58422.2023.10198920.

[245] *VRChat*. https://hello.vrchat.com/. Accessed: March 18, 2025. 2025.

[246] Hans Wackernagel. "Ordinary Kriging". In: *Multivariate Geostatistics: An Introduction with Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 74–81. ISBN: 978-3-662-03098-1. DOI: 10.1007/978-3-662-03098-1_11.

[247] *Grid Constraints and Data Center Investment: Microsoft's Strategic Shift*. Dublin, Ireland: Bisnow, 2025, pp. 12–28.

[248] Jialin Wang et al. "Effect of Frame Rate on User Experience, Performance, and Simulator Sickness in Virtual Reality". In: *IEEE Transactions on Visualization and Computer Graphics* 29.5 (2023), pp. 2478–2488. DOI: 10.1109/TVCG.2023.3247057.

[249] Xiao Sophia Wang, Arvind Krishnamurthy, and David Wetherall. “Speeding up Web Page Loads with Shandian”. In: *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. Santa Clara, CA: USENIX Association, 2016, pp. 109–122. ISBN: 978-1-931971-29-4.

[250] Xiao Sophia Wang et al. “Demystifying Page Load Performance with WProf”. In: *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. Lombard, IL: USENIX, 2013, pp. 473–485. ISBN: 978-1-931971-00-3.

[251] Ranysha Ware et al. “Beyond Jain’s Fairness Index: Setting the Bar For The Deployment of Congestion Control Algorithms”. In: *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*. 2019, pp. 17–24.

[252] *Web Almanac*. https://almanac.httparchive.org/en/2022/. 2023.

[253] *WebPageTest - Website Performance and Optimization Test*. https://www.webpagetest.org/. Accessed: 2019-09-10. 2019.

[254] Mark West and Chew Han Ei. *Reading in the mobile era: A study of mobile reading in developing countries*. UNESCO, 2014.

[255] Keith Winstein and Hari Balakrishnan. “Tcp ex machina: Computer-generated congestion control”. In: *ACM SIGCOMM Computer Communication Review* 43.4 (2013), pp. 123–134.

[256] Keith Winstein, Anirudh Sivaraman, and Hari Balakrishnan. “Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks”. In: *Proc. of NSDI*. Lombard, IL, Apr. 2013, pp. 459–471. ISBN: 978-1-931971-00-3.

[257] Keith Winstein, Anirudh Sivaraman, Hari Balakrishnan, et al. “Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks.” In: *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation (NSDI 13)*. Lombard, IL, 2013.

[258] *World Development Report 2016: Digital Dividends.* https://www.worldbank.org/en/publication/wdr2016. Accessed: 2021-12-20. 2016.

[259] WorldData.info. *WorldData.info: The World in Numbers.* Accessed: 2025-04-08. 2025.

[260] Tim Wu. "Network neutrality, broadband discrimination". In: *J. on Telecomm. & High Tech. L.* 2 (2003), p. 141.

[261] Dongzhu Xu et al. "Understanding operational 5G: A first measurement study on its coverage, performance and energy consumption". In: *Proc. ACM SIGCOMM.* 2020, pp. 479–494.

[262] Tan Xu, Bo Han, and Feng Qian. "Analyzing viewport prediction under different VR interactions". In: *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies.* CoNEXT '19. Orlando, Florida: Association for Computing Machinery, 2019, pp. 165–171. ISBN: 9781450369985. DOI: 10.1145/3359989.3365413.

[263] Elias Yaacoub and Mohamed-Slim Alouini. "A Key 6G Challenge and Opportunity - Connecting the Remaining 4 Billions: A Survey on Rural Connectivity". In: (Nov. 2019). DOI: 10.36227/techrxiv.10253336.v1.

[264] Francis Y Yan et al. "Pantheon: the training ground for Internet congestion-control research". In: *2018 USENIX Annual Technical Conference (USENIX ATC 18).* 2018, pp. 731–743.

[265] Ofir Zafrir et al. "Prune Once for All: Sparse Pre-Trained Language Models". In: *International Conference on Learning Representations (ICLR).* 2021.

[266] Yasir Zaki et al. "Adaptive congestion control for unpredictable cellular networks". In: *Proc. ACM SIGCOMM.* 2015, pp. 509–522.

[267] Yasir Zaki et al. "Dissecting Web Latency in Ghana". In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada: Association for Computing Machinery, 2014, pp. 241–248. ISBN: 9781450332132. DOI: 10.1145/2663716.2663748.

[268] Menglei Zhang et al. "Transport layer performance in 5G mmWave cellular". In: *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2016, pp. 730–735.

[269] Menglei Zhang et al. "Will TCP work in mmWave 5G cellular networks?" In: *IEEE Communications Magazine* 57.1 (2019), pp. 65–71.

[270] Xiaoqing Zhu and Rong Pan. "NADA: A Unified Congestion Control Scheme for Low-Latency Interactive Video". In: *2013 20th International Packet Video Workshop*. 2013, pp. 1–8. DOI: 10.1109/PV.2013.6691448.

[271] Tommaso Zugno et al. "Implementation of a Spatial Channel Model for ns-3". In: *Proceedings of the 2020 Workshop on ns-3*. 2020, pp. 49–56.