University of California
Santa Barbara

Autonomous and Predictive Systems to Enhance the Performance of Resilient Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Vivek Adarsh

Committee in charge:

Professor Elizabeth Belding, Chair
Professor Arpit Gupta
Professor Trinabh Gupta

September 2022

The Dissertation of Vivek Adarsh is approved.

Professor Arpit Gupta

Professor Trinabh Gupta

Professor Elizabeth Belding, Committee Chair

July 2022

Autonomous and Predictive Systems to Enhance
the Performance of Resilient Networks

Dedication here

Acknowledgements

I believe Ph.D. is all about the journey and not about the destination. I was extremely fortunate to have an incredible journey with amazing people who have been my collaborators, confidantes, friends, mentors, and family. I will try my best to thank every one of them, but if I forget someone below, know that I am lucky to have known them, and they made my journey all the more meaningful.

This work is based on co-authored publications with Elizabeth Belding, Michael Nekrasov, Udit Paul, Paul Schmitt, Tarun Mangla, Alex Ermakov, Esther Showalter, Arpit Gupta, Ellen Zegura, and Morgan Vigil-Hayes. Thanks to my coworkers and lab-mates, Paul Schmitt, Morgan Vigil-Hayes, Esther Showalter, and Mai El-Sherief, for their support and the endless hours of fruitful discussion. Thank you for your professionalism and friendship. A special thanks to Michael Nekrasov and Udit Paul for the fantastic ideas, endless hours of fieldwork, and being great travel and research companions as we worked together in environments where things never went as planned. Research and our road trips would not have been fun without both of you.

I wish to express my greatest gratitude to my committee members Elizabeth Belding, Arpit Gupta, and Trinabh Gupta, for guiding me throughout my degree. I interacted with Arpit significantly during my Ph.D. as a researcher interested in his work and during his networking courses.

Thank you to my research advisor, Elizabeth Belding. Her advice and wisdom have been invaluable. I admire her extreme patience during our meetings and her approach to understanding complex ideas from fundamentals. If not these amazing skills, I hope I learned how always to stay humble and listen to collaborators, from her. Elizabeth has been a constant source of inspiration throughout my journey and

allowed me to chase crazy ideas purely because they were interesting. (For instance, camping out at a Starbucks outside a Lakers' game just to collect cellular congestion data.) A significant portion of my personal and professional transformation as a person over the last half-decade is credited to her teachings. Being an international student comes with unique challenges, but Elizabeth's unwavering support during tough times was incomparable. I could not have asked for a better research advisor, teacher, and mentor. Lastly, thank you for the opportunities to explore networks in far-flung regions that I could not have imagined studying before being a part of the Moment lab.

None of this research would have been possible without many academic and industry collaborators. In particular, thank you to Anu Mercian, Diman Zad-Tootaghaj, and Puneet Sharma for your mentorship and guidance during two wonderful summers at Hewlett Packard Labs. Thank you for letting me work on far-fetched ideas that somehow turned into implementable projects. Your support was crucial in ensuring that we met the deliverables timely. Thank you to Jerrold Baca for your time and resource investment. It was a pleasure to work with you. Thanks to many local partners, including the Southern California Tribal Chairmen's Association's Tribal Digital Village program, the Iipay Nation of Santa Ysabel, Santa Clara Pueblo, and Ohkay Owingeh.

I am the person today because of my family. My parents have given endless support and have been exemplary role models for how to be a decent human. My dad instilled the value of working hard and pushed me always to dream big. His countless sacrifices and prioritizing my choices over his own enabled me to reach this place. My mom's care and support allowed me to focus on my education while she took care of far more critical issues. I may not say it enough – but I am incredibly grateful to be their son.

Curriculum Vitæ

Vivek Adarsh

Contact

2152A Harold Frank Hall
UC Santa Barbara
California 93106

vivek@cs.ucsb.edu
https://vivekadarsh.com
https://linkedin.com/in/vivekadarsh

Education

University of California, Santa Barbara **July 2022**
Doctor of Philosophy (Ph.D.), Computer Science

University of California, Santa Barbara **June 2018**
Master of Science (M.S.), Computer Science

University of Pune **June 2016**
Bachelor of Technology (B.Tech), Electrical Engineering

Professional Experience

Research Associate Intern - Hewlett Packard Labs **Jun 2021 - Sep 2021**
Networking and Distributed Systems Lab
Project - SmartFabric: AI-Acceleration on the fly using DPUs

Graduate Research Assistant **Mar 2017 - Jul 2022**
University of California, Santa Barbara
Department of Computer Science

Research Associate Intern - Hewlett Packard Labs **Jun 2020 - Sep 2020**
Networked Systems Group
Project - SARAS: Intelligent Edge Acceleration using SmartNICs

Software Engineer Intern - LogMeIn Inc. **Jun 2018 - Sep 2018**
Predicting cloud infrastructure failures and automated troubleshooting using DeepLearning models.

Research Intern - EIGEN Technology **Jun 2015 - Aug 2015**
Application of DSR and LEACH protocols on wireless sensor nodes (SenseNuts).

Research Intern - NVIDIA Singapore **Jan 2015 - Apr 2015**
Automated extraction and conversion of unstructured video metadata to structured data (U2S) using parallel computing (Pascal architecture).

Awards & Recognition

- 2^{nd} Runner-up, New Venture Startup Competition **2021**
- Won the *Best in Class* competition among 400 interns at Hewlett Packard Enterprise (international competition) **2020**

Press & Media Coverage

Millions of Americans can't get broadband because of a faulty FCC map. There's a fix. By Shara Tibken, CNET. February 19^{th} 2021.(CNET has the world's highest readership among Web news sites, with more than 200 million readers per month.)
– Available here: `https://www.cnet.com/features/millions-of-americans-cant-get-broadband-because-of-a-faulty-fcc-map-theres-a-fix/`

Networking and Distributed Systems Lab's intern wins Best in Class for summer project. By Curt Hopkins, Hewlett Packard Labs. September 24^{th} 2020.
– Available here: `https://community.hpe.com/t5/Advancing-Life-Work/Networking-and-Distributed-Systems-Lab-s-intern-wins-Best-in/ba-p/7102384#`

Patents & Inventions

Directing Queries to Worker Nodes of a Cluster of a Container Orchestration Platform Distributed across a Host System and a Hardware Accelerator of the Host System
Vivek Adarsh, Diman Zad Tootaghaj, Anu Mercian, Puneet Sharma.
Hewlett Packard Enterprise. **US Patent Application – 17/222160**

Refereed Publications

[1] **AKIDA: Accelerating In-Network, Transient Compute Elasticity using SmartNICs**
Vivek Adarsh, Diman Zad Tootaghaj, Anu Mercian, Puneet Sharma.
ACM/IEEE Symposium on Edge Computing (SEC) [Under Submission]

[2] **Characterizing Internet Access and Quality Inequities in California M-Lab Measurements**
Udit Paul, Jiamo Liu, David Farias-llerenas, **Vivek Adarsh**, Arpit Gupta, Elizabeth Belding.
ACM Conference on Computing Systems and Sustainable Societies **(COMPASS '22)**

[3] **Too Late for Playback: Estimation of Video Stream Quality in Rural and Urban Contexts**
Vivek Adarsh, Michael Nekrasov, Udit Paul, Alex Ermakov, Arpit Gupta, Morgan Vigil-Hayes, Ellen Zegura, Elizabeth Belding.
Proceedings of the Passive and Active Measurement Conference **(PAM '21)**

[4] **Estimation of Congestion from Cellular Walled Gardens using Passive Measurements**
Vivek Adarsh, Michael Nekrasov, Udit Paul, Elizabeth Belding.
IEEE Transactions on Mobile Computing.

[5] **Coverage is Not Binary: Quantifying Mobile Broadband Quality in Urban, Rural, and Tribal Contexts**
Vivek Adarsh, Michael Nekrasov, Udit Paul, Tarun Mangla, Arpit Gupta, Morgan Vigil-Hayes, Ellen Zegura, Elizabeth Belding.
Proceedings of the IEEE 30th International Conference on Computer Communications and Networks (**ICCCN '21)**

[6] **A Tale of Three Datasets: Towards Characterizing Mobile Broadband Access in the United States**
Tarun Mangla, Esther Showalter, **Vivek Adarsh**, Kipp Jones, Morgan Vigil-Hayes, Elizabeth Belding, Ellen Zegura.
Communications of the ACM **(CACM)**

[7] **Packet-level Overload Estimation in LTE Networks using Passive Measurements**
Vivek Adarsh, Michael Nekrasov, Ellen Zegura, Elizabeth Belding.
Proceedings of the Internet Measurement Conference **(IMC '19)**.

[8] **#Outage: Detecting Power and Communication Outages from Social Networks**
Udit Paul, Alex Ermakov, Michael Nekrasov, **Vivek Adarsh**, Elizabeth Belding.
Proceedings of the World Wide Web Conference **(WWW '20)**.

[9] **MPTCP Performance over Heterogeneous Subpaths**
Vivek Adarsh, Paul Schmitt, Elizabeth Belding.
IEEE 28th International Conference on Computer Communication and Networks **(ICCCN '19)**.

[10] **Evaluating LTE Coverage and Quality from an Unmanned Aircraft System**
Michael Nekrasov, **Vivek Adarsh**, Udit Paul, Esther Showalter, Ellen Zegura, Morgan Vigil-Hayes and Elizabeth Belding.
IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems **(MASS '19)**.

Invited Talks

Decoding Cellular Walled Gardens
Google Inc. (Cloud and NetInfra Teams) February 2020

Professional Services

- Technical Program Committee, 2021
  ACM SIGCOMM Artifact Evaluation Committee
- External Reviewer, IEEE Wireless Communications and Networking Conference (WCNC) 2021
- External Reviewer, IEEE Wireless Communications Letters Journal 2020
- Judge, SB Hacks VII (UCSB's Annual Hackathon Competition) 2021

Teaching

- Introduction to Computer Networks (CMPSC 176A) — Spring 2019
- Computer Networking (CMPSC 176B) — Winter 2019
- Introduction to Computer Networks (CMPSC 176A) — Fall 2018
- Introduction to Computer Networks (CMPSC 176A) — Spring 2018
- Computer Networking (CMPSC 176B) — Winter 2018
- Introduction to Computer Networks (CMPSC 176A) — Fall 2017
- Advanced Topics in Internet Computing (CMPSC 176C) — Spring 2017

Abstract

Autonomous and Predictive Systems to Enhance
the Performance of Resilient Networks

by

Vivek Adarsh

Over half the planet's population currently has access to the Internet in some form. Consequently, Internet usage has increased exponentially on mobile and fixed networks. In particular, video traffic now accounts for 80% of all Internet traffic, largely thanks to the rise of streaming platforms and video content from social media. With this tremendous growth, it becomes challenging to maintain consistent and high-quality service to all connected users. This is particularly true in the case of video sessions, which demand relatively consistent, high volume bandwidth to maintain high-quality sessions for end-users.

However, the presence of connectivity does not necessarily equate to usable service. Even in well-provisioned deployments, network resiliency can be an issue. Our own assessment of multiple networks across the US indicates that there is a wide disparity between reported coverage and actual usability. Numerous studies, including ours, indicate that LTE penetration in rural areas is far less than what is typically reported by providers and much less prevalent and lower quality than that in urban areas. Similarly, for fixed broadband, studies have observed highly variable median speeds across the globe, including South America, the Middle East, and Africa. Within the US itself, broadband speeds vary considerably as well. Finally, networks have to support multitudes of end-user devices (e.g., mobile phones, laptops, desktops) and a wide variety of application types, many with challenging and strict delivery requirements. As a result, ensuring high-quality, low-variability

service quality under variable conditions is exceptionally challenging.

This dissertation is aimed at tackling this challenge. We assess how existing technologies and standards function in complex environments: lack of deterministic pathways for Internet connectivity, presence of congestion and other resource constraints, and temporal or spatial variability in network performance. Based on our extensive analysis of real-world conditions, we produce cellular and broadband systems designed to bridge the technological gap for end-users in challenging network conditions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Accessibility to the Internet has seen hypergrowth in the past decade, both on mobile and fixed networks. Unsurprisingly, over half the planet's population has access to the Internet in some form, shape, or manner[1]. However, within an expanding user base, a significant trend has emerged. Internet video traffic has grown fourfold in the past half a decade. By the end of 2022, Internet video will be 82% of all consumer Internet traffic [2], largely due to the rise of streaming platforms and video content from social media. To achieve necessary performance and resiliency, a planned and collaborative suite of solutions must address challenges across the networking stack to keep pace with global connectivity demand and provide a seamless user experience.

With this tremendous growth in adoption, it becomes challenging to maintain consistent and high-quality service to all connected users. *Connectivity may not necessarily equate to usable service.* Since the introduction of the Connecting America: The National Broadband Plan in 2010, increased efforts have been made to bring connectivity to areas where it previously was not [3]. However, even in well-provisioned deployments, network resiliency could be an issue. Our studies indicate a wide dis-

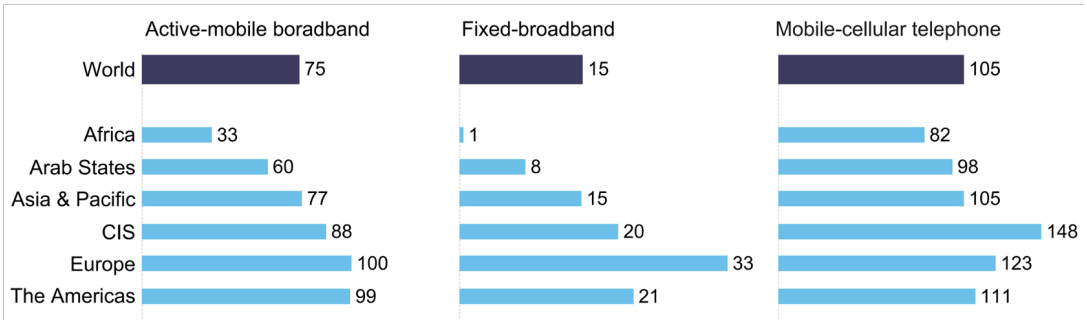


Figure 1.1: Mobile-cellular, fixed-broadband and active-mobile broadband telephone subscriptions per 100 inhabitants, 2020 [Source: ITU Facts & Figures 2020]

parity between reported coverage and actual usability [4, 5]. Figure 1.1 shows the number of subscriptions for every 100 people globally. We learn that combining North and South America, the ratio for cellular subscriptions exceeds 1. Stated otherwise, it means individuals may hold multiple cellular subscriptions. This figure indicates that the number of subscriptions is artificially bolstered by people carrying multiple SIM cards to obtain service on multiple service providers depending on time and place. Further, we observe that networks serving rural areas and developing regions continue to significantly lag behind the rest of the world [6]. Networks that do offer coverage in such resource-limited areas are often poor-quality and offer spotty, unpredictable coverage[7].

To provide appreciable service quality, it is imperative first to assess the existing service quality and network performance to optimize and strategically inform future resilient networks' design. Mobile broadband standards such as LTE and 5G are frequently used to connect the edges of access networks to the Internet. Assessing such networks in terms of coverage and performance remains a significant challenge. Service providers frequently use generous propagation models to assess coverage, often overstating availability and not accounting for quality and usability[8, 9, 10].

Availability of network connectivity does not typically guarantee the lack of fre-

quent disruptions either. In rural locations, for example, under-provisioned networks face frequent outages and provide a service quality that can fall far short of what users paid for[4]. In addition, natural disasters can cripple communication infrastructure and severely damage well-provisioned networks when most needed. After Hurricane Harvey, 4.1% of cell sites across the coasts of Texas and Louisiana were inoperable [11]. Hurricane Irma disrupted 27% of towers in affected areas of Florida [12] and downed 55% of Puerto Rico's cellular towers [12]. Hurricane Maria disabled 95% of cellular sites in Puerto Rico and 76% in the Virgin Islands [13]. The impact of these disasters sometimes lasts months or years. Infrastructure damage to power and communication networks in Puerto Rico took more than 11 months to restore [13]. Additionally, in a well-provisioned region, sudden escalation in traffic demand from user equipment (UE) can occur during large gatherings (e.g., street festivals, protests). Prior work has demonstrated that even in areas that cellular providers claim are well-covered, persistent over-usage due to insufficient capacity can exist [14]. To remedy this disparity between reported coverage and actual usability, individual users, watchdog groups, and government agencies need tools to verify whether a network serves customers adequately. Mobile network operators (MNOs) already use emerging technologies, such as Unmanned Aerial Systems (UAS), to assess network damage and performance. The work in this dissertation further expands capabilities by proposing systems of automated network quality assessment and systems assessing network congestion while maintaining scalability.

While instantaneous examination of service quality is necessary, it is not adequate to ensure a seamless user experience for subscribers. Network providers require deep inspection into the end-user experiences, typically a function of the application layer performance, to allocate appropriate resources for smoothly rendering application-specific content on multitudes of user devices. This examination of Quality of Ex-

perience (QoE) is resource-intensive, requires observability into active monitoring devices (unlike passive subscribers' devices), and has scalability limitations. To that end, we need accurate predictive models to improve device-agnostic user experience – integral to the design of future resilient networks.

In this thesis, we focus on *measurements and characterization* of real-world networks and subsequent *system designs* based on our *predictive analysis* to improve connectivity in the measured contexts.

1.1 Thesis Statement

This dissertation shows that:

Typically, network conditions and service quality in mobile and fixed networks vary non-uniformly across space and time. To achieve global connectivity and normalize the end-user experience, we must empirically assess these networks and use those findings to build automated systems that inform the design of resilient networks.

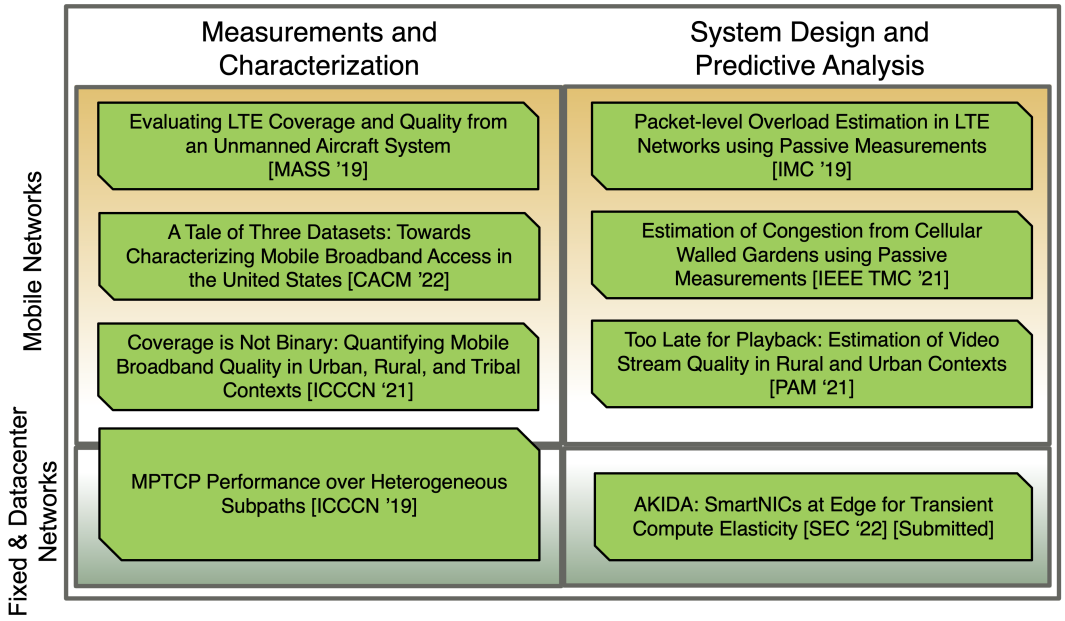


Figure 1.2: Dissertation overview.

We measure and analyze multiple real-world, resource-limited networks with temporally and spatially varying performance and design new predictive systems to mitigate the challenges we observe.

The structure of this dissertation is summarized in Figure 1.2 and is organized as follows. The dissertation is divided into two parts. We first measure and characterize network topologies in challenging environments. We leverage our findings to inform system designs to predict network performance at scale.

1.2 Key Contributions

In this section, we discuss our broad contributions and their implications based on the outline shown in Figure 1.2.

1.2.1 Measurements and Characterization

We perform measurement and analysis on two access technologies: (i) mobile broadband and (ii) fixed broadband.

- ***Mobile broadband***: We collect cellular network measurements in order to characterize performance and identify under-performing aspects of the infrastructure under study.

  Contributions: In our first dataset, we evaluate the accuracy of simple, lightweight spectrum sensing devices for LTE RSRP measurements compared to that of expensive solutions. We then assess the gradation within which similar RSRP measurements are obtained from complex equipment currently employed. To do so, we initiate a wardriving campaign to collect ground and air LTE signal strength (RSRP) measurements in two regions in New Mexico over a period of five days. We employ six unique RF sensing methodologies in our analysis. These ground measurements are uniquely helpful in contrasting the efficacy of each measurement technique. We then show that these devices can be mounted on Unmanned Aircraft Systems (UAS) to more rapidly and easily measure coverage across wider geographic regions. Our results show that the low-cost aerial measurement techniques have 72% accuracy relative to the ground readings of user equipment and fall within one quality gradation 98% of the time. The dataset above was also used to quantitatively examine the LTE coverage

disagreement among existing datasets collected using different methodologies. We find that existing datasets display the most divergence when compared with each other in rural and tribal areas.

Our second dataset focuses on the comprehensive assessment of network performance of various LTE service providers. The dataset consists of QoS and QoE datapoints that are spatially and temporally varying. We collect 16 datasets from 12 locations across the Southwestern U.S. Eight of those datasets are collected from rural locations with sparse cellular deployment. Six of the eight locations are under American Indian Tribal sovereignty jurisdiction, while two are from non-tribal rural regions. In addition, we collect another eight datasets from four urban locations in California. For each urban location, we collect two datasets: one during a large event or gathering, in which we expect cellular network congestion to occur, and a second during typical operating conditions. Hence, our traces are broadly classified into three categories: rural and tribal, congested urban, and baseline urban. Our analysis confirms that the performance of LTE networks in tribal and rural areas (that we study) is typically worse than even heavily congested urban networks. More specifically, in the regions that we study, LTE networks in under-provisioned (tribal/rural) areas have 9× poorer video streaming quality, 10× higher video start-up delay. They undergo more than 10× the number of resolution switches and lead to more than 2× slower Web browsing experience compared to urban deployments. Despite identical LTE carrier subscription plans, we show that throughput and latency are 11× and 3× worse in tribal and rural locations.

- ***Fixed broadband***: We investigate traces collected in broadband networks to examine topology characteristics and transport protocols' performance.

Contributions: In order to understand the path characteristics of the Multipath TCP (MPTCP) protocol, we gather network traces from Web downloads with varying object sizes. The study's goal was to examine the performance of the default MPTCP scheduler. For this set of experiments, we deploy two MPTCP-enabled machines on a popular cloud service provider. Our testbed consists of a server located in Virginia and a client that is situated in California. The client communicates with the server over the Internet through two different wired interfaces, each connected to a different subnet to maintain isolation of routes. We collect network traces to compute throughput, loss rate, and latency measurements, along with page load times of the websites. This study shows how heterogeneous paths can adversely affect MPTCP performance, especially when one path is lossy.

1.2.2 System Design and Predictive Analysis

We design novel systems capable of accurately inferring network and application performance for providers to improve the Quality of Service (QoS) and Quality of Experience (QoE) for end-users.

- ***Mobile broadband***: Through our measurement campaigns, we find that detecting overload and congestion is challenging without embedding active monitoring devices into the cellular network. Ideally, public entities should be able to assess the overload, congestion, and operational status/usability of cellular base stations. Further, they should be able to accomplish this without relying on the cooperation of a cellular provider. To address this critical need, we propose a novel solution to infer overload and congestion in LTE networks based on messages broadcast by the eNodeB. Further, estimation of end-user experi-

ence as a function of network utilization (overload/congestion) and deployment sparseness is difficult to achieve, resource-intensive, and requires unscalable approaches. To that end, we design novel, scalable systems to estimate user experience.

Contributions: To study network overload and congestion using only passive measurements, we develop Lumos, a data analysis platform capable of quantifying overload in eNodeBs; and Edain, a networking monitoring suite that automates the collection of Quality of Service (QoS) and Quality of Experience (QoE) metrics. Through the analysis of multiple message types, we draw clear comparisons between instances of high network utilization and typical operating conditions for several eNodeBs (LTE base stations). Further, we evaluate performance differences incurred from overload-driven congestion through QoS and QoE metrics assessment. Our results indicate that eNodeBs demonstrate measurable performance differences indicative of overload conditions and network congestion.

Given that cellular deployment is often guided by economic demand, concentrating deployment in urban areas and leaving economically marginalized and sparsely populated regions under-served, the wireless nature of LTE networks necessitates that QoE be evaluated in multiple locations per base station as factors such as signal availability may have significant spatial variation. Based on our observations that quality of service (QoS) metrics are less time and resource-intensive to collect, we investigate how QoS can be used to infer QoE in LTE networks. Using an extensive, novel dataset representing a variety of network conditions, we build several state-of-the-art predictive models for scalable video QoE inference. We demonstrate that our models can accurately predict rebuffer-

ing events and resolution switching more than 80% of the time, despite the dataset exhibiting vastly different QoS and QoE profiles for various network conditions we study. We also illustrate that our classifiers have a high degree of generalizability across multiple videos from a vast array of genres. Finally, through an ablation study, we highlight the importance of low-cost QoS measurements such as reference signal received power (RSRP) and throughput in QoE inference.

- ***Fixed broadband***: Our analysis of applications on fixed broadband and data center traffic enables us to identify unique challenges facing such networks and inform the design of scalable systems using domain-specific accelerators at the network's edge.

  Contributions: The growth of edge-computing systems is driven by solutions and applications requiring high-performance and low-latency video conferencing and streaming services. The adoption of serverless frameworks to process applications at the edge has increased significantly. However, provisioning for additional computing needs on a transient basis for sudden workload spikes, or *transient elasticity*, is non-trivial. Scaling serverless functions at the edge poses critical challenges. Service-level agreement (SLA) violations are typically frequent in such scenarios. Since SLA violations carry severe penalties, one common way to eliminate violations is to over-allocate resources preemptively. This solution leads to the under-utilization of expensive resources. Meanwhile, SmartNICs (smart network interface cards) have gained popularity to offload various network functions and provide real-time, line-rate computing at scale.

  Our study proposes AKIDA, a new architecture that strategically harvests the untapped compute capacity of the SmartNICs to offload transient workload

spikes, thereby reducing the SLA violations. Usage of this untapped compute capacity is more favorable than adding and deploying additional servers, as SmartNICs are economically and operationally more desirable. AKIDA is a low-cost and scalable platform that orchestrates seamless offloading of serverless workloads to the SmartNICs at the network edge, eliminating the need for pre-allocating expensive compute power and over-utilization of host servers. Our system evaluation shows that SLA violations can be reduced by 20% for certain workloads.

1.3 Broader Impacts

In addition to peer-reviewed publications and presentations to academics, this work was impactful to the larger community.

- We have partnered with multiple Native American communities to assist in the mapping, evaluating, and deploying LTE and TVWS technologies. We partnered with Southern California Tribal Digital Village [15], New Mexico Santa Clara Pueblo, and New Mexico Ohkay Owingeh tribe to develop technology in line with the needs of these communities.

- We provided outreach and training at Santa Clara Pueblo, Ohkay Owingeh, and Northern New Mexico Community College, where we taught local children and adults basic principles of internet performance measurements as well as LTE space measurement basics. Further, we held several one-on-one discussions of our work with partners and community members.

- This work has generated considerable interest in the industry. In 2020, we gave an invited presentation to the NetInfra team at Google. Since 2021, our

investigation of video streaming applications in rural areas [5] led an industry partner, ViaSat, to sponsor the continuation of a similar project at our research facility that supports a full-time doctoral student.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows. In Chapter 2 we present the background of our research, including a description of our field studies and challenges encountered during measurement campaigns. In Chapter 3 – Chapter 5 we characterize cellular network performance and user behaviors in various settings. In Chapter 6 includes an in-depth look at the performance of the Multipath TCP protocol evaluated under real-world conditions. Chapter 7 provides the motivation and philosophy of our system design for Chapter 8 – Chapter 11. Specifically, in Chapter 8 – Chapter 9 we discuss the development of two robust and scalable systems we built to infer overload and congestion in cellular networks. Chapter 10 presents our design of network models to easily accurately predict user experience for video streaming applications from inexpensive QoS measurements. Next, we demonstrate the efficacy of AKIDA, a serverless system designed to process applications in-network using SmartNICs, in Chapter 11. Lastly, Chapter 12 discusses our findings and concludes this dissertation.

Chapter 2

Research Background

Accessible and usable Internet service is crucial as a fundamental human right [16]. Globally, Internet access has become increasingly available through mobile broadband LTE cellular networks [17, 18]. Although more than 4 billion users worldwide benefit from Internet access provided by LTE, economic opportunism often drives LTE and other broadband technology expansion, concentrating deployment in densely populated urban centers. Economically marginalized and sparsely populated rural areas often remain under-served [19, 20, 21], expanding the digital divide between those with usable Internet access and those without [22, 23]. In the United States, for example, rural tribal regions suffer from the poorest LTE coverage [24]. When crises occur, like the worldwide COVID-19 pandemic, the digital divide heightens inequalities for rural communities [25, 26]. As education, work, and social interactions transition to online environments, those with poor broadband access have no way to participate [27].

To bridge the divide, communities require accurate assessments of existing access and usability gaps. Government entities or third-party crowd-sourced data firms frequently quantify LTE mobile broadband access using coverage maps that represent

signal availability as binary. In the United States, the Federal Communications Commission (FCC) releases data based on self-reporting by Mobile Network Operators (MNOs) [28]. This data influences incentive programs for cellular deployment in rural regions [29, 30]. However, due to the reliance on self-reported data and the binary nature of coverage reports, researchers repeatedly cast doubts on the data's accuracy [31, 10, 32]. In addition to government sources, third-party data firms [33, 34] gather more nuanced coverage estimation, including signal strength and throughput measurements. Due to the crowd-sourced nature of these data sets, data is primarily available for urban centers and transit corridors and is, consequently, sparse in rural and tribal regions. New advances in coverage mapping using automated sensors, including Unmanned Aircraft Systems, show promise as a means of effectively mapping rural areas [35].

In this dissertation, we study the usage and performance of various networking solutions across space and time. We leverage our trace captures and application-related performance metrics to identify network usage and challenges related to user experience and design systems based on our findings. The remainder of this chapter discusses the background of our fieldwork and a brief description of the characteristics and challenges of working in diverse environments.

2.1 Field Work

We conduct fieldwork in multiple locations around the US, representing varying cellular deployment and network usage levels. We first collect measurements from the United States based LTE and Wi-Fi networks. To study the variation in Multipath TCP (MPTCP) scheduling decisions made over different wireless access technologies, we measure packet round-trip times (RTTs) for the Tranco top 10K websites [36] over

different interfaces: Wi-Fi and LTE. RTT is a crucial parameter in network performance; the default MPTCP scheduler makes path selection decisions that rely heavily on the RTT of each path. Different interfaces can produce vastly different RTTs for the same web servers, partly due to server replication within CDNs. Next, server reachability is critical in assessing whether MPTCP can be implemented in a broad range of scenarios. Additionally, it helps us understand the importance of selecting the primary interface. As the first step in our study, we perform DNS resolution of the Tranco top 10K websites [36] on each of the available interfaces (Wi-Fi and LTE) on the monitoring device using the tethered phone. Web servers can resolve to different IPs over different interfaces. That is because resolution depends on how the ISP routes the request to return the address of the desired content delivery network (CDN). For instance, cellular operators will likely embed web servers and CDNs within their core network to provide faster response times to user web requests. Omitting websites that do not resolve, the resulting sample size after DNS resolution is 9756 and 9638 for Wi-Fi and LTE, respectively, with an overlap of 58.74% in IP addresses. We then conduct RTT tests on the obtained web servers using Hping3. Hping3 uses TCP packets to ping the servers.

Next, we capture traces of cellular control channel messages, performance probe traffic, data packet captures, and QoS/QoE data over mobile broadband networks.

- ***Santa Fe, New Mexico***: To study the association of mobile network coverage, performance, and user experience, we undertake an extensive measurement campaign to collect network traces from eight locations in the Southwestern United States for four major telecom operators: AT&T, Sprint, T-Mobile, and Verizon. In this work, we compare rural regions to their urban counterparts. For our rural datasets, we partnered with rural and tribal communities in New

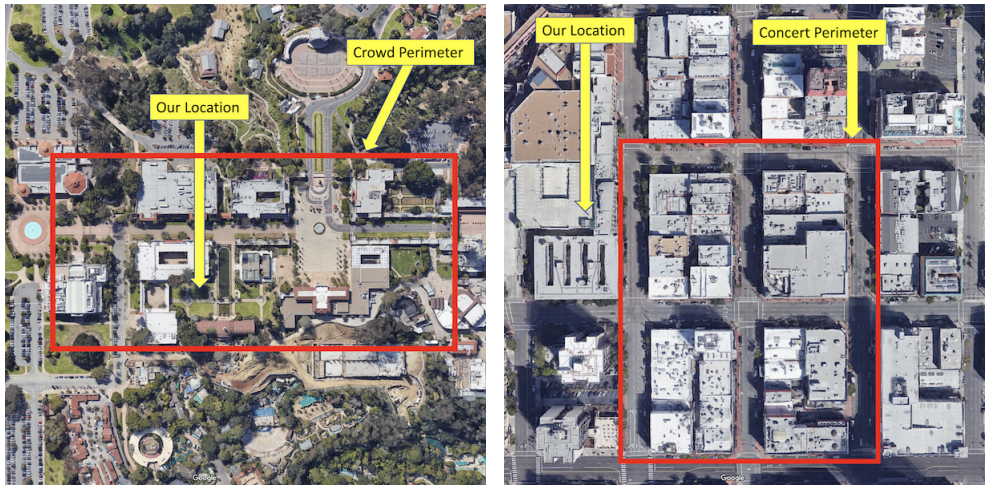
Mexico because: (i) the state has the second-highest percentage of American Indian population (8.77%) in the United States, just behind Alaska; (ii) it is home to 23 Indian tribes - nineteen Pueblos, three Apache tribes, and the Navajo Nation, had the highest poverty rate in the nation at 21.9% [37]; (iii) anecdotal evidence indicated poor network usability and inadequate coverage in the target locations. At each location, we run extensive tests, capturing signal coverage, network QoS, and QoE for two common applications: Web browsing and on-demand streaming video. The designation of each location as tribal or rural is based on information gathered from the Census Bureau [38]. We collect throughput, latency, packet loss, and Reference Signal Received Power (RSRP) measurements for QoS metrics. QoE measurements include start-up delay, video quality, resolution switches, rebuffering ratio, buffer size, and page load times. We run our measurement suite on four Lenovo ThinkPad W550s laptops, each tethered to its own Motorola G7 Power (Android 9) via USB to measure cellular performance. The measurement campaign ran from May 28^{th} – June 1^{st} 2019. We collect and analyze over 32.7 million LTE packets through these measurement campaigns.

- ***St. Patrick's Day, San Diego***:

  To study overload conditions in LTE networks, we propose a novel solution to infer overload based on messages broadcast by the eNodeB (LTE base stations). Through the analysis of multiple message types, we draw clear comparisons between instances of high network utilization and typical operating conditions for several eNodeBs. To test our proposed solution, we identify times and locations in which we anticipate cellular overload, capture traces, and then compare network performance in those traces with baselines captured in the exact loca-

tion during normal operating conditions (when no network overload is likely to occur). We select spaces that are anticipated to have large gatherings but are unlikely to be provisioned for large crowds (i.e., city streets as opposed to stadiums that typically have sufficient network capacity to handle crowds). Overall, our dataset consists of over 3.2 million frames, with data collection that lasts for a cumulative duration of about 5.2 hours. While it is not possible to compute the exact number of UEs in the vicinity due to the lack of international mobile subscriber identity (IMSI) number in broadcast messages for security reasons, measuring the number of temporary unique UE IDs (`uniqueUeID`) in RRC Connection Requests allows us to estimate the number of active UEs present nearby. The measurement setup consists of an LTE-enabled USRP tethered to a Lenovo ThinkPad W550s running Lumos, our scalable platform that infers network overload and congestion.

- **St. Patrick's Day (SPD)** We collect cellular traces during the St. Patrick's Day parade adjacent to Balboa Park in San Diego, CA [39]. The parade was held on Saturday, March 16^{th} 2019, beginning at 10:00 AM and ending around noon, while the public fair lasted through 3:30 PM. We physically position our data collection devices within the crowd to better assess the eNodeBs serving this particular region, as shown in (Figure 2.1(a)). The total duration of data collection is about 76 minutes, which resulted in over 1.1 million LTE frames. We observe 27,349 `uniqueUeIDs`.
- **St. Patrick's Day Baseline (SPD_base)** As a point of comparison for the SPD dataset, we again gather LTE traces from the same location, from 8 PM to 9 PM on Tuesday, March 26^{th} 2019. Collection in the evening on a weekday helped us avoid unexpected large gatherings in the park's many



((a)) **SPD dataset**.
Balboa Park, San Diego, CA

((b)) **CSR dataset**.
Downtown San Diego, CA

Figure 2.1: Google aerial map of experimental datasets - SPD/CSR.

venues while still capturing local nightlife activity. Compared to the *SPD* dataset, we expect this dataset to exhibit low levels of overload, acting as a baseline for the location. Indeed, we see about 6,992 `uniqueUeIDs`. We collect a little over 275K LTE frames in 65 minutes.

– **ShamROCK Concert (CSR)** We collect traces from the ShamROCK concert in the downtown area of San Diego [40] on March 16^{th} 2019. The event started at 7:00 PM and lasted until midnight. We collect 113 minutes (~1.7 million LTE frames) of traces during this period. This event/location combination (as shown in Figure 2.1(b)) was selected because we anticipated that the amount of cellular traffic during the event would well exceed the typical traffic load. Because this location (city streets) does not typically have large crowds, we expect there to be network overload during a large event. This dataset contains 42,433 `uniqueUeIDs`.

– **ShamROCK Concert Baseline (CSR_base)** As a baseline to the *CSR* dataset,

we capture additional traces (~135K LTE frames) in the same location from 9:30 PM to 10:30 PM on March 26^{th} 2019, when the number of pedestrians and amount of vehicular traffic was more representative of normal operating hours. We detect only 3,338 `uniqueUeIDs` during this data capture.

- ***Suburbs, San Diego***:

  Similar to SPD and CSR datasets and their experimental setup, we undertake a separate, extensive measurement campaign to assess overload and congestion in LTE networks.

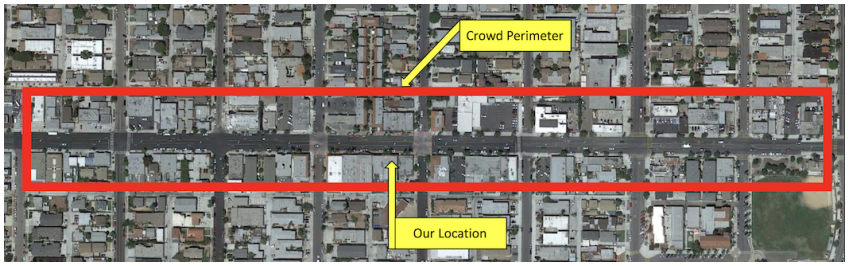
  - **Adams Street Fair (ADM)** We gather LTE traces during the 38th annual Adams Avenue Street Fair in the Normal Heights neighborhood of San Diego [41]. The street fair was held on Sunday, September 22^{nd} 2019, beginning at 10:00 AM and concluding at 6:00 PM. We physically position our networking gear in a cafe on the same street as the fair (Adams Avenue) to better assess the eNodeBs serving this particular region, as shown in (Figure 2.2(a)). The total duration of data collection is 129 minutes, which resulted in over 1.63 million LTE frames. In addition, we observe 59,084 `uniqueUeIDs`.
  - **Adams Street Fair Baseline (ADM_base)** As a point of comparison for the ADM dataset, we collect LTE traces from the same location, from 7:00 AM to 9:00 AM on Saturday, September 28^{th} 2019. Collection early in the morning on the weekend helped us to avoid unexpected large gatherings in the neighborhood while still capturing the activity of local residences and businesses. Compared to the *ADM* dataset, we expect this dataset to exhibit low levels of overload, acting as a baseline for the location. Indeed,

we see about 5,307 uniqueUeIDs. We collect a little over 316K frames in 57 minutes.

- **Concert WaterFront (CWF)** We collected traces from the CRSSD music festival concert at the Waterfront Park in downtown San Diego [42] on Sunday, September 29^{th} 2019. We monitor the event between 4:00 PM and 7:00 PM. In total, we collect 126 minutes (1.89 million frames) of traces during this period. This event/location combination (as shown in Figure 2.2(b)) was selected because we anticipated that the amount of cellular traffic during the event would well exceed the typical traffic load. Over this two-day event, there were an estimated 15,000 attendees. Because the waterfront does not typically have large crowds, we expect there to be network overload during a large event. This dataset contains 69,728 uniqueUeIDs.
- **Concert WaterFront Baseline (CWF_base)** As a baseline to the *CWF* dataset, we capture additional traces (442K frames) in the same location on Monday, September 30^{th} 2019, from 10 to 11 AM, when the number of pedestrians and amount of vehicular traffic was more representative of normal operating hours. We detect only 7,478 uniqueUeIDs during this data capture.

- ***The Palace Of Fine Arts, San Francisco***: We gather this dataset from San Francisco Bay Area to assess congestion and overload conditions. The experimental setup remains identical to SPD and CSR.

  - ***AI Summit (AIS)***: We gather traces at the AI Summit, held on September 25^{th} 2019 at the Palace of Fine Arts in San Francisco [43] between 10:00 AM to 3:00 PM. The event attracted more than 6000 participants hosted



((a)) **ADM dataset**. Adams Avenue, San Diego

((b)) **CWF dataset**. Waterfront Park, San Diego

((c)) **AIS dataset**. Palace of Fine Arts, San Francisco

Figure 2.2: Google aerial map of experimental datasets - ADM/CWF/AIS.

within the confinements of the venue. Because of the size and tech-centric nature of the event, which requires participants to be digitally connected, we anticipated cellular congestion. Even though the venue-provided Wi-Fi coverage, anecdotal evidence suggests that a major fraction of participants were on cellular service - vibrantly exhibited through severe congestion on all of our test and personal mobile devices. This behavior could be attributed to the need to proactively login to Wi-Fi through the dedicated conference app, which would inevitably require downloading via cellular data. After parsing our dataset, we observe about 2.34 million LTE dataframes collected over 149 minutes. In addition, we identify 111,404 `uniqueUeIDs`.

– ***AI Summit Baseline (AIS_base)***: To establish a baseline for network per-

formance in *AIS*, we run our network monitoring suite at the same location. The venue was closed when the measurements were taken at 9:00 PM on September 26^{th} 2019. Hence, we collect traces from the parking lot, roughly 30 meters away from our previous placement (i.e., *AIS*). To minimize any disparity arising from a slight location change, we ensure that all our devices monitor and connect to the same cell towers that were present in *AIS* (this is achieved by matching the `CellID` parameter). The tests ran for over 65 minutes and collected over 396K frames. This dataset contains 6,089 `uniqueUeIDs`.

2.2 Challenges in Data Collection

This section presents our experience analyzing cellular and Internet connectivity and designing systems based on our analysis. This summary is based on traffic analysis and our anecdotal experience while spending time in the communities we have studied.

The underlying economics and purchasing power of a limited customer base has typically been the limiting factor in securing robust connectivity in rural areas. From the service provider's standpoint, there is no motivation to invest in expensive infrastructure that will not likely result in revenue that outweighs the capital and operating costs. As a result, deployment is sparse, and service quality is poor for most users. Due to a lopsided cost-to-benefit ratio, our anecdotal information reveals that residents prefer wired-landline phones to cellular subscriptions. In tribal regions that in our investigation fare worse than rural regions, cellular services are often unavailable, spotty, or of poor quality at best. As a result, tribal community members often travel tens of miles to secure acceptable service quality from providers. This

observation indicates that most activities that heavily rely on broadband access are most likely performed asynchronously, despite the Internet's promise to enable synchronous operations in a fast, reliable manner. Further, communities on tribal lands have comparatively lower per-capita income [44] than other US populations. The lack of and poor quality technological infrastructure and limited access to financial resources present seemingly insurmountable obstacles to communication. Yet, in precisely these circumstances, communication is most critical [45].

It is equally challenging to collect network traces from periods of high network consumption that lead to overload conditions to detect real-world congestion occurrences without a precise assessment of the prior provisioning by cellular service providers. Before collecting LTE packets in San Diego, our research team undertook many resource-intensive measurement campaigns in Los Angeles (stadiums and concerts) and San Francisco (packed events downtown). We speculate that the economic demand at these specific hot spots led the service providers to proactively over-provision the network with little to no service disruptions. The lack of congested traffic, despite being unrepresentative of most locations, resulted in unusable datasets for us. In addition, measurement campaigns that employ off-the-shelf radio devices during events can seemingly appear as a security threat to law enforcement agencies. Consequently, several of our planned measurement campaigns were cut short at the request of law enforcement officers (for instance, during concerts). Such adverse encounters lead to wasted opportunities for data collection. In other scenarios, where a security clearance was mandatory, the long processing times often did not offer an adequate temporal cushion for the campaigns to materialize. In our experience, gathering measurements from crowded events (a pre-requisite for a potential congested network) pose significant logistical challenges.

2.3 Discussion and Conclusion

Global connectivity is a requirement in today's society. Consequently, we must address the issues that come with pushing the boundaries of Internet accessibility further. While the global expansion of LTE/5G and fixed Internet access in recent years has been one of humanity's greatest scientific triumphs, a significant portion of the world's population has yet to be reached. If service is provided, it must be of usable quality. This thesis looks at the particular issues that communication infrastructure faces in various network conditions, including both rural and urban settings. For the scenarios we study, we identify areas for improvement and create robust systems to address them.

Our effort in this thesis is to extend the state-of-art networking infrastructure to enable faster, reliable, and accessible Internet to individuals who do not have it and to areas where it is critically needed. Through our studies, we better understand the end-user experience within different network conditions across the continental United States. We create network systems based on our findings to assist in bringing the next billion people online and improve the experience for those still hampered by poor connectivity.

Part I

Network Characterization

Chapter 3

Evaluating LTE Coverage and Quality from an Unmanned Aircraft System

3.1 Introduction

Billions of users worldwide benefit from high-speed internet access provided by LTE. However, economic incentives often drive LTE and other broadband technology expansion, concentrating deployment in populated urban areas. Economically marginalized and sparsely populated rural areas remain underserved [19]. In the United States, for example, rural tribal regions suffer from the poorest LTE coverage [24]. Even when cellular providers claim coverage, the poor signal quality can limit achievable download data rates far below the mobile broadband threshold, defined by the U.S. Federal Communications Commission (FCC) as a median speed of 10 Mbps [24].

For underserved regions in the United States, the FCC has instituted incentive programs to offset provider infrastructure deployment costs [29, 30]. These programs determine the bounds of existing coverage and identify coverage deficiencies by semi-

annually collecting network connectivity reports from commercial network operators. Every operator that owns cellular network facilities in the United States participates in data collection by submitting a Form 477 [46]. The reported coverage area consists of geo-polygons using an operator-defined methodology. Based on this data, the FCC allocates subsidies to incentivize commercial coverage in underserved regions and verifies compliance.

The FCC publicly releases annual Broadband Deployment Reports (e.g. [24]), as well as shapefiles for each operator that indicate geographic coverage areas [46]. However, researchers challenge their accuracy [31, 10, 32]; for example, Meinrath et al. examined a public dataset of speed tests collected by the Measurement Lab [47, 48] and demonstrated that broadband access in Pennsylvania is much lower than claimed in the report. This inaccurate over-reporting can be attributed to the proprietary and often generous propagation models used by network operators [9]. To validate the actual state of mobile broadband access, we need publicly controlled methods for measuring coverage areas and signal quality, particularly in underserved regions.

To audit provider-reported coverage claims, third parties undertake independent measurement efforts. While the concept of "coverage" remains imprecise [49], network parameters such as the received signal strength (in terms of Reference Signal Received Power (RSRP)) are typically used to estimate the extent of network availability. Popular public crowdsourcing platforms, such as CellMapper[50] or OpenSignal [33], collect measurements from network users and calculate cellular coverage and signal quality. Data from crowdsourced efforts provide information over time and for a wide range of devices, but these data cluster around significant transportation arteries, omitting communities outside of these areas. An alternative collection strategy employs specialized equipment with dedicated users. For example, wardriving typically involves physically navigating rugged terrain in remote

areas to record on-the-ground measurements [51]. This method enables greater control over the measurement process and the geographic scope but scales poorly due to considerable time investment and labor costs.

Because existing strategies suffer from the abovementioned drawbacks, we need alternative solutions for measuring LTE coverage and signal quality. An ideal method should enable quick assessment using measurements of the RF landscape throughout large areas (on the order of square miles), even if hard to access. In addition, new strategies should provide scalability for equipment and human resources. Based on these criteria, off-the-shelf software-defined radios offer a viable solution. However, because SDRs can cost anywhere between tens of dollars to a few thousand dollars, it is crucial to study the relationship between the precision of readings and the cost of the equipment to determine whether more affordable equipment will suffice.

Unmanned Aircraft Systems (UAS) can carry payloads while maintaining appreciable flight times to reduce the human effort of spectrum scans. The availability of low-cost, programmable, highly agile unmanned aerial vehicles has spurred interest in employing aerial RF sensing for cellular coverage mapping [52, 53]. UASs enable coverage for large geographic areas, which may be costly, complex, or impossible to cover on foot or the land. Network operators, such as Verizon, already employ UASs for visually inspecting equipment after natural disasters [54]. Extending UAS capabilities to include signal measurements is an active area of interest for various wireless applications [55, 56]. These extensions could further enable uses for scalable rural cellular coverage mapping as well as post-disaster recovery efforts. Small form-factor Software Defined Radios (SDR) with high sensitivity, such as the RTL-SDR (RTL2832U chipset with an Elonics E4000 Tuner), are proving increasingly useful for LTE applications [57, 58, 59]. This SDR is ideally suited for UAS application. However, the high altitude of UAS flight relative to the ground poses challenges to the

efficacy of these approaches. As antennas on LTE towers are provisioned for ground transmission, the RF radiation pattern picked up at high altitudes may not reflect signal quality on the ground [60].

This chapter assesses the accuracy of a low-cost, small form-factor RTL-SDR for sensing LTE eNodeB signal strength over a wide area through integration with an off-the-shelf quadcopter UAS. To do so, we first compare the reading accuracy of this airborne sensor with commonly used hardware for ground-based wardriving approaches (i.e., a spectrum analyzer and a USRP). Further, because no existing studies systematically examine the effect of altitude on signal strength measurements, we fly the UAS at varying altitudes across multiple locations and examine how aerial signal sensing can be aligned to ground-level measures. Because minimal previous research compares observed signal strength between measurements collected by the user equipment (UEs) (i.e., smartphones, tablets, and hotspots) and UASs, we deployed four cellular devices on the ground, each collecting measurements from different cellular networks, and compared these measurements over the same geographic area to those collected by the RTL-SDR on the UAS. We look to the UE measurements as “ground truth" because the UE readings capture examples of the actual coverage and performance a user in the given location would experience with UE.

Our findings reveal that the simple RTL-SDR has comparable accuracy to expensive solutions and can estimate quality within one gradation of accuracy compared to the user equipment. Further, we show that these devices can be mounted on a UAS to measure coverage across wider geographic regions more rapidly. Our results show that the low-cost aerial measurement techniques have 72% accuracy relative to the ground readings of user equipment and fall within one quality gradation 98% of the time. Our findings, taken together, offer a detailed look at the efficacy of low-cost,

public-controlled aerial coverage and quality sensing.

This chapter proceeds as follows: Section 3.2 overviews related work. Section 3.3 explains the methodology and the corresponding datasets, while our analysis and results are presented in Section 3.4. Section 3.5 discusses impact of this study and future work, and Section 11.7 concludes the chapter.

3.2 Related Work

RF Spectrum Sensing with SDRs: Previous studies involving wide-scale cellular sensing include analysis of GSM pollution [61] and propagation model verification for LTE signals [62]. In one study, an SDR was mounted on a UAS to navigate the flight path by LTE location signalling [63]. In contrast, our study uses only passive sensing with a very lightweight SDR to discover ground truth signal strength readings. We show that low-cost equipment detects LTE availability to produce a coverage map using RSRP measurements that aligns with the ground truth UE measurements. In our study, we adopt a wide-scale television white space sensing approach used by Saeed et al. [64], but we adjust for LTE instead of TV frequencies.

RF Spectrum Sensing with UASs: Considerable prior work has focused on identifying the application of UASs for cellular networks. Batistatos et al. [52] study the variation in LTE signal strength and SINR for both an underserved rural area and an urban center. A UAS connected to an existing LTE network monitored the LTE signals in various altitudes [53]. The authors found that at 60m to 100m above the ground, LTE coverage probability climbs to 90%, and the received power gains 18 dB for the ground level. However, this work did not compare measurements taken from the aerial platform to those taken from a ground-level UE.

Lin et al. [60] shed light on the applicability and performance of mobile network connectivity to low-altitude UASs by analyzing downlink channel indicators, such as RSRP. While the study in [65] examines the variation of RSRP, RSRQ, and SNR throughout a drive-by style campaign in an urban university campus using a passive monitoring device, little prior work has explored the effective measurement of received signal strength using SDRs accompanied by an adequate validation from UE readings.

LTE Performance Measurement with RSRP:

Estimation of received signal strength plays a vital role in many control plane operations, including inter- and intra- eNodeB handovers [66, 67, 68, 69, 70]. Precise detection of RSRP plays a crucial role in these handovers and several diagnostic methods in LTE networks. For instance, Anas et al. [71] evaluate the performance of RSRP handovers in LTE. They observe that a handover margin of 2dB to 6dB (RSRP) leads to an optimal number of handovers without sacrificing much uplink SINR (for a specific range of user velocity). The effect of RSRP measurement bandwidth on the accuracy of handovers is studied in [72, 70]. From a telecom provider's perspective, this suggests a need for up-to-date, accurate RSRP space maps for improving service quality.

Several prior works examine the relationships between RSRP, RSRQ, and SINR [73, 74, 75, 76], but little work explores the correlation between passive monitoring of LTE channels and ground UE readings. In [77], the researchers examine the viability of deploying LTE connectivity using UASs in rural areas. Their results indicate that the coverage outage level increases from 4.2% at an altitude of 1.5 m to 51.7% at 120 m under full load conditions. Another study analyzed a set of live network measurements conducted with an LTE scanner attached to an airborne UAV [78]. The findings

suggest improved radio clearance as the UAV increases altitude. The increase in the average number of detected cells as altitude increases corroborates these findings. Our study offers the first look into conducting reliable RSRP measurements with UAS using low-cost off-the-shelf SDRs to our knowledge.

3.3 System Overview and Methodology

We collected ground and air measurements in two regions in Rio Arriba County, New Mexico, for five days, beginning May 28, 2019. For each region, we obtained permission to drive through residential areas and fly a UAS equipped with a sensor.

This section describes the six unique RF sensing methodologies employed in our analysis. Like all wardriving studies, our work is necessarily limited in scale. However, these ground measurements are uniquely helpful for contrasting the efficacy of each measurement technique. In all cases, our methodology is generalizable. Figure 3.1 shows images of many of our sensing set-ups.

Method 1: Ground-Driven User Equipment (UE) Sensing:

In our wardriving campaign, we record signal strength readings from four Motorola G7 Power (XT1955-5) phones, each running Android Pie (9.0.0). We collect measurements using the Network Monitor application [79]. An external GlobalSat BU-353-S4 GPS connected to an Ubuntu Lenovo ThinkPad laptop gathered geolocation measurements, which we matched to the appropriate ground measurement by timestamp. We outfitted each phone with a SIM card from one of the four top cellular providers in the region: Verizon, T-Mobile, AT&T, and Sprint. The phones recorded signal strength every 10 seconds while we drove at speeds less than 10 miles per hour through the areas of study.

a. UE b. Ground Measurement Kit

c. UAS d. Stationary box

Figure 3.1: Sensing equipment.

Method 2: Ground-Driven Spectrum Analyzer:

We gathered measurements on LTE channel center frequencies with a high-precision Keysight N9340b spectrum analyzer (SA) using a ham radio antenna capable of sensing signals up to 3 GHz. The SA was transported inside the measurement vehicle while the antenna was magnetically mounted to the roof.

Method 3: Ground-Driven USRP:

We collected center frequency readings with an Ettus Research USRP B200, a versatile software-defined radio widely used for LTE and TV frequency experimentation and sensing. The USRP measured the same LTE channel center frequencies as the SA through a ham radio antenna placed beside the identical SA antenna.

Method 4: Ground-Driven RTL-SDR:

We also collected center frequency readings with a NooElec RTL-SDR RTL2832U and Elonics E4000 Tuner, an inexpensive software-defined radio operating at 55MHz-1100MHz and 1500MHz-2300MHz ranges. The RTL measured LTE channel center frequencies through a ham radio antenna placed beside the identical SA and USRP antennas.

Unlike the USRP B200, which, when inside a transportable case, is bulkier and more expensive, the RTL-SDR is low-cost and protected in a smaller form factor. This specific model of RTL-SDR covers most LTE frequencies and is simple to equip onto a UAS or deploy at a static site for long-term monitoring. This ground-transported RTL-SDR is a comparison point for the UAS and longitudinal sensing experiments described in subsequent subsections.

Methods 5: Aerial Sensing Platform:

Our UAS consisted of a DJI Matrice 100 quadcopter, as shown in Figure 3.1(c). The UAS collected signal strength readings via a NooElec RTL-SDR (the same model used for ground measurements) connected to a Raspberry Pi 2—Model B onboard computer via USB. The location of the UAS was recorded from the Matrice 100 onboard GPS, sampling at a rate of 50 Hz and using a UART connection to the Pi.

Horizontal Coverage Mapping. In one set of experiments, we manually flew the UAS at varying speeds and elevations (to clear obstacles and keep the UAS in line

of sight) to map coverage. We attempted to cover the same areas as the UE and ground measurements. UAS measurements occurred the same day as the other data collection for each geographic region, but sometimes several hours apart.

Vertical Experiments. To investigate the impact of elevation on signal strength measurements, we performed four sets of vertical-only flights. Each set of flights was conducted in a different geographic region of our measurement area. During each vertical flight, the UAS was raised by 10ft increments approximately every 15 seconds to 100ft. It was then raised in 20ft increments roughly every 15 seconds to 400ft (the maximum FAA non-exempt altitude limit).

Method 6: Stationary Box:

Because continuous monitoring in an area can be costly in terms of equipment and manpower, coverage mapping is typically completed via sampling over a short time frame. For example, in our ground sensing driving campaign, we take all samples over a maximum of one hour for each unique location. As part of our study, we seek to verify that this one-shot sensing method is appropriate for estimating long-term spectrum availability.

We, therefore, measure spectrum occupation over time in a single location to monitor changes. We enclosed a NooElec RTL-SDR (the same model as is utilized by the ground measurements and UAS) run by a Raspberry Pi 3 B+ in a weather-proof case with the stock antenna on top of the case, shown in Figure 3.1(d). Over two days, the RTL continuously iterated through a pre-programmed list of all 20 known LTE frequencies for the four network providers in the area and recorded signal strength readings for each frequency every three seconds.

This method monitors the stability of the RTL-SDR measurements over time. It can indicate the appropriate flight time necessary to generate a consistent measure

of signal quality in an area. While this data is not generalizable geographically, it provides insight into the precision of signal strength reading from an RTL-SDR.

LTE Channel Selection by Provider:

Before data collection, we compiled a list of LTE cellular frequencies in use by the top providers in the area. This compilation was needed for every sensing method other than the UEs, which pulled the active frequencies automatically for their respective provider. We compiled this list using two complementary processes. First, on each UE, we ran CellMapper [50], an Android application that allowed us to query the active frequencies detected by the device for the corresponding LTE provider. We supplemented this list with a scan using a spare Ettus Research USRP B200, equipped with a wideband LTE dipole antenna [80], connected to a Lenovo ThinkPad laptop running srsLTE [81]. Using srsLTE we performed a scan of all possible LTE frequencies operated in the United States and appended to our list any frequencies not previously discovered. Since UEs choose the strongest frequency to communicate with a nearby base station, we could locate other frequencies available from nearby cells, which the UEs would not use at our test sites but could jump to intermittently. As we moved between regions, we added all newly detected frequencies to the list scanned by all sensors.

The resulting list contained 22 frequencies in the area served by the four providers. Because the NooElec RTL-SDR is limited to 2300 MHz, two of the detected frequencies (2628.8 MHz and 2648.6 MHz) were outside the range frequencies we could sense on the Ground RTL, Stationary Box, or UAS and are dropped from our analysis.

	UE	UAS	Spectrum Analyzer	USRP	RTL-SDR
UE	1152	-	-	-	-
UAS	305	812	-	-	-
Spectrum Analyzer	131	53	1199	-	-
USRP	131	53	1199	1199	-
RTL-SDR	131	53	1199	1199	1199

Table 3.1: Number of overlapping geographic bins by signal collection method.

3.4 Analysis

3.4.1 Accuracy of Data Collection Methods

Preparing data for geographical analysis

Because multiple devices and personnel participated in data collection, the data was not sampled at the same timestamp or precise GPS location for all methods. For example, the ground sensors and the UAS passed over the same residential area but may not have covered the same 1-meter GPS coordinate due to road availability. To accurately compare data collected by different methods, we first aggregate data into geographical bins of three decimal places of GPS accuracy, approximately 110 square meters in area. Then, for each method and each set of readings on different LTE frequencies, we take the mean across all the signal strength values that fall into that geographic bin.

The data collected by the UEs included only the network provider (AT&T, Sprint, T-Mobile, and Verizon) and not the frequency on which the UE was operating. To compare this to the other data collection methods, which report frequency instead of the network, we first map each frequency to the corresponding network provider using the frequency list described in Section 3.3. For each network provider and geographic bin, we then select the frequency with the strongest signal strength and set

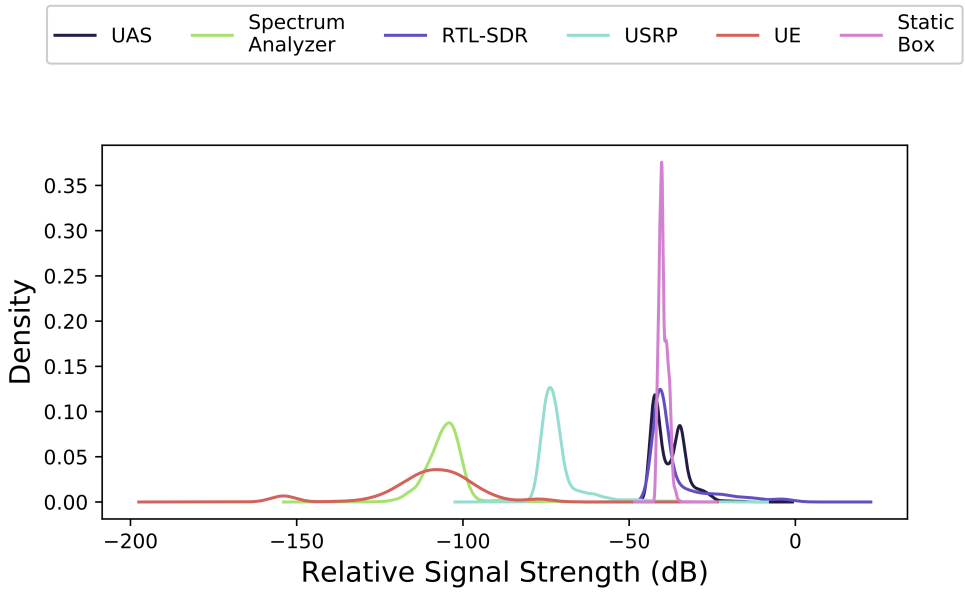


Figure 3.2: Kernel density estimation of original data by signal collection method.

that as the signal strength for the provider in that bin. This method resulted in 2,637 unique 110m^2 geographic bins. Not every area was sampled by every method. The resulting overlap between methods and geographic area is summarized in Table 3.1.

Transforming Raw Signal Readings

The UEs, spectrum analyzer, RTL-SDR modules (including the one mounted on the UAS and in the stationary monitoring box), and the USRP all report signal strengths on varying scales. We show the original distribution of the relative signal strengths for each collection method in Figure 3.2. As we can see, while the distributions have similar normal-like peaks, the offsets and width do not match.

While the spectrum analyzer outputs dBm, the other devices report relative signal strengths. As we are interested in the end user's experience, we first need to transform the raw relative signal strength readings to match the UEs before comparing data collection methods.

To do so, we first perform a min/max normalization on each method, as shown in Equation 3.1, where $\vec{O}$ is the original data, $m \in \{spectrum\ analyzer, USRP, RTL, UAS\}$ is the method and $\vec{N}$ is the normalized data.

$$\vec{N}_m = \frac{\vec{O}_m - \min(\vec{O}_m)}{\max(\vec{O}_m) - \min(\vec{O}_m)} \tag{3.1}$$

Next, we offset and scale the other methods to align them with the signals received by the UE. To do this, we randomly selected 50% of our data as a training set. On this training set for each method, we find an offset x_m and scaling factor a_m to minimize Equation 3.2, where n_m and n_{ue} are measures taken from the same geographic bin and cellular network provider. If a method does not have a matching UE measurement, it is omitted from the sum.

$$\min_{x_m, a_m} \left(\sum_{n_m \in \vec{N}_m} [a_m(n_m + x_m) - n_{ue}]^2 \right) \tag{3.2}$$

Finally, we scale and offset *all of our data* for every method other than the UE by x_m and a_m, and scale back to the readings of the UE as expressed in Equation 3.3, where $\vec{T}_m$ is the resulting transformed data for each method. We can now compare signal strength readings to one another and the UE by transforming the data. The resulting transformed distributions are shown in Figure 3.3. We use this transformation for the rest of our analysis to report signal strength values in dBm.

$$\vec{T}_m = \left[a_m \left(\vec{N}_m + x_m \right) \right] \left(\max(\vec{N_{ue}}) - \min(\vec{N_{ue}}) + \min(\vec{N_{ue}}) \right) \tag{3.3}$$

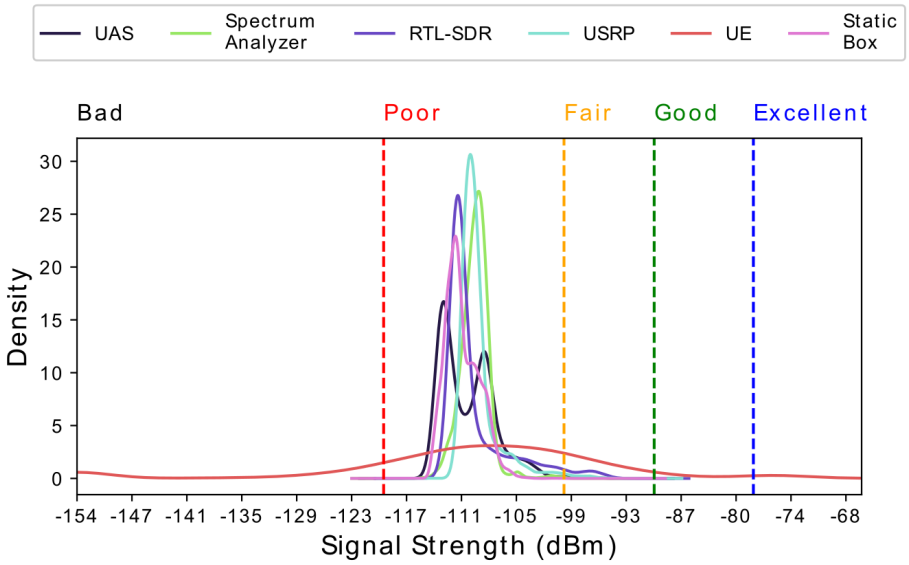


Figure 3.3: Kernel density estimation of transformed distributions by signal collection method.

Estimating signal strength

We computed the Pearson correlation on each method pair. We found only a weak linear relationship between the collection methods and the readings from the UEs, even after transforming the data. As signal strength can vary, even between different UE device makes and models, we categorize the level of signal quality rather than predicting the exact signal strength a UE would receive in an area by dividing the signal strength levels into five groups based on criteria in Table 3.2. While there is no standard for defining what LTE signal strength corresponds to what quality, we

Signal Quality	Range	Color
Bad	<120 dBm	Black
Poor	-120 to -111 dBm	Red
Fair	-111 to -105 dBm	Orange
Good	-105 to -90 dBm	Green
Excellent	>90 dBm	Blue

Table 3.2: Categorization of signal strength into signal quality bins.

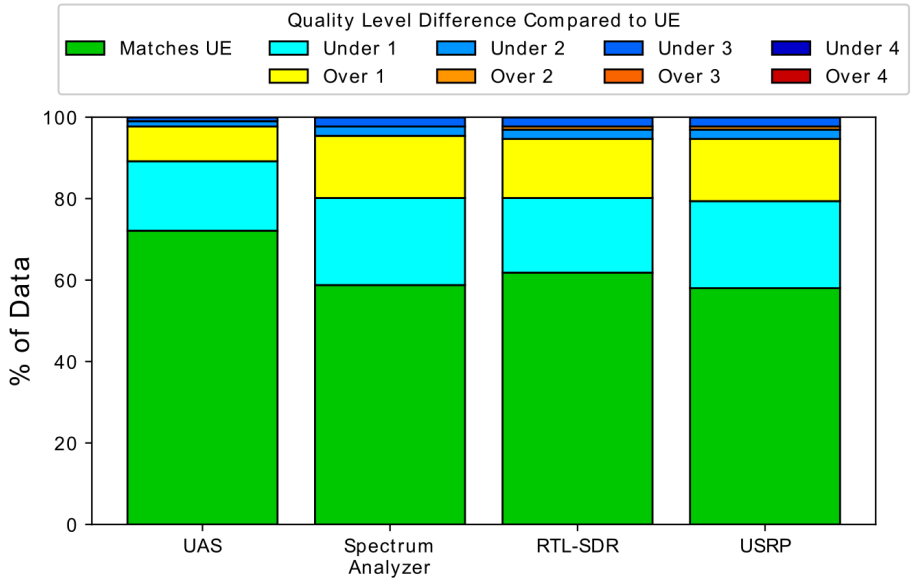


Figure 3.4: Accuracy of signal collection methods as compared to the UEs.

model our criteria after those suggested by SignalBooster [82].

Based on this categorization, we compare how each method sorted the signal strength readings across the geographical bins, using the UE as ground truth. We summarize our results in Figure 3.4. The UAS was closely aligned with the UE, matching values for 72% of the geographic bins. When allowing for one signal quality of discrepancy (for instance, a method stating a signal was Fair when the UE labeled it as Poor), all methods had over a 95% accuracy, with the UAS again leading with a 98% accuracy. A notable result was that error was skewed towards under-predicting the received signal strength. Accounting for this bias when estimating UE reception would improve accuracy further.

3.4.2 Longitudinal Analysis

From the stationary radio (introduced in Section 3.3), we received 684,096 readings over two days. To measure the relative stability of signal strength readings,

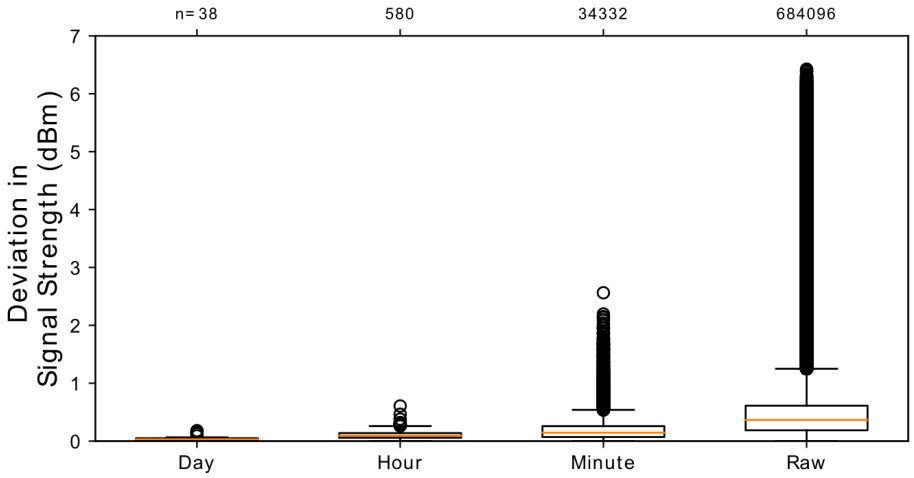


Figure 3.5: Distribution of deviation from mean signal strength of all monitored LTE frequencies.

we calculated the deviation from the mean of the corresponding frequency for each reading. To determine the stability across different time scales, we re-sampled the data over multiple time scales (1 minute, 1 hour, 1 day), averaged the intermediate readings, and re-computed the deviation of each sample. The resulting time-series is shown in Figure 3.5.

As expected, raw readings (with a sampling frequency of 3 Hz) fluctuated considerably, with a total range of 80dBm and the majority of fluctuations < 7 dBm from the mean. When comparing minutes, most of the reading was < 3 dbm from the mean. When comparing hour to hour, most signals deviate < 1 dBm from the mean. Comparing two days of data across all frequencies, most signals did not deviate.

We analyzed the distribution of hour-to-hour signal strengths across the 20 monitored LTE frequencies, as shown in Figure 3.6. Most readings across the two-day time span fell within 1 dBm of each other. The most significant change in signal strengths between two hours was observed on 739.0 MHz (utilized by AT&T) and 1967.5 MHz (used by Verizon), which exhibited 7dBm changes.

The end-user is most impacted by the signal strength of the frequency chosen by

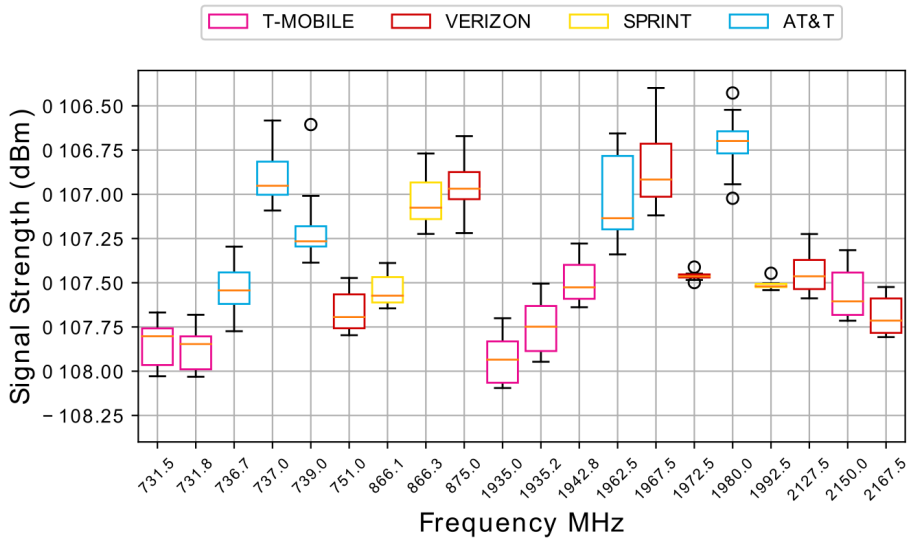


Figure 3.6: Distribution of signal strength by frequency over period of observation, averaged across one hour windows.

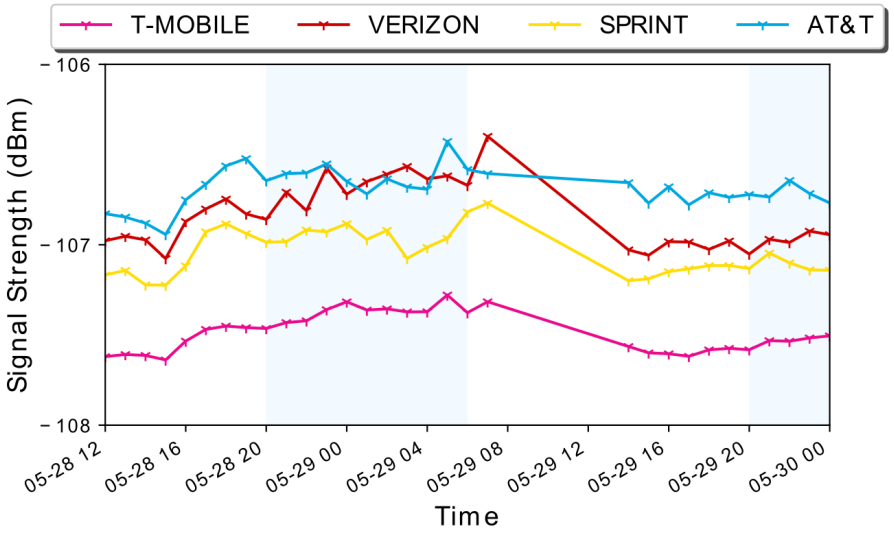


Figure 3.7: Signal strength over time by provider gathered by stationary box. Night hours are shaded in blue.

the UE. We also examine the hour-by-hour change in network signal strength. For each operator, we choose the frequency with the maximum average signal strength for every hour time window. We present the results in Figure 3.7. While there is a slight improvement in signal strength during nighttime hours, for each network, the

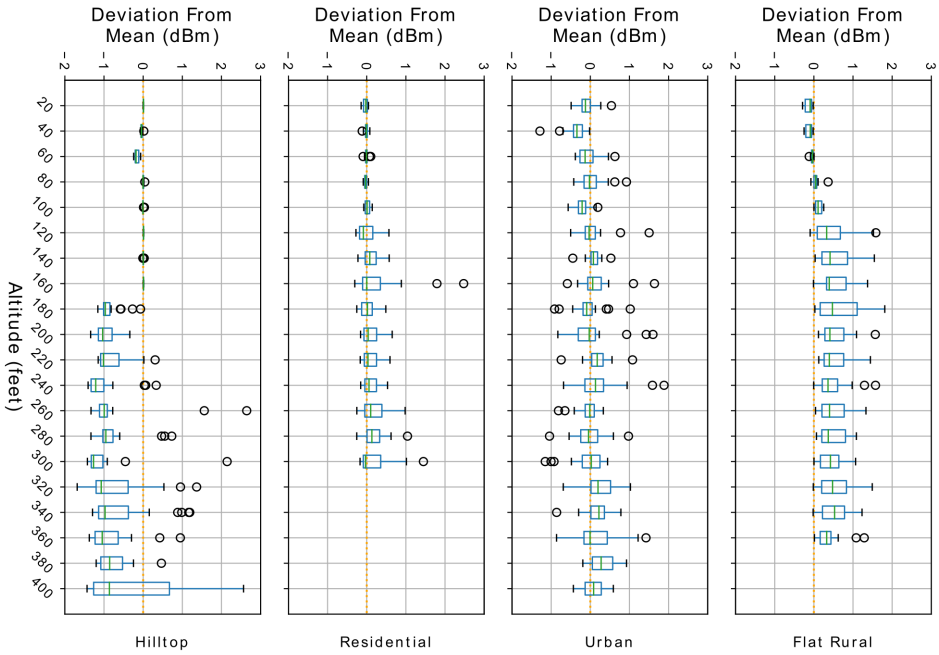


Figure 3.8: Deviation of signal strength from mean.

total hour-to-hour fluctuations in signal strength are minimal.

3.4.3 Impact of Altitude

To analyze the impact of sampling altitude on signal strength, we executed multiple vertical flights in four different locations, as described in Section 3.3. In our analysis, we keep the four locations separated and examine how signal strength at each LTE frequency varies with altitude from the ground. To compare frequencies, we calculate the deviation of each signal strength measurement from the mean of that frequency at each location. We then group altitudes into 20-foot bins and examine the distributions of altitudes across those bins at each of the four locations. We present the results in Figure 3.8.

Our results show that signal strength variation can be quite dependent on location. The first and third locations, *Flat Rural* and *Residential* respectively, were located

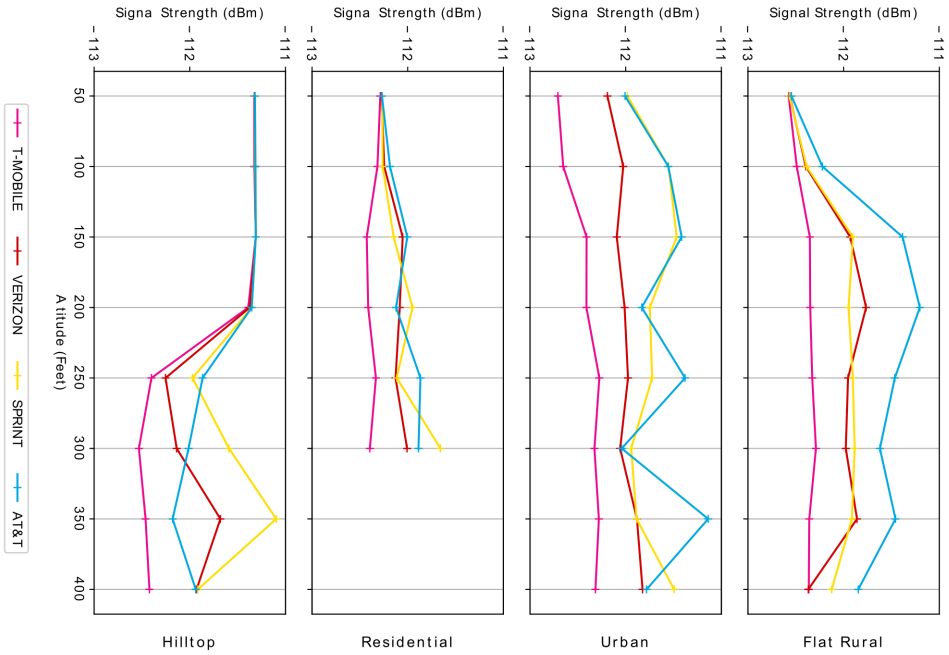


Figure 3.9: Signal strength change by altitude and network.

level with a wide area of flat terrain surrounded by low hills. At these locations, the vertical UAS flight showed an overall increase in signal strength as altitude increased across LTE frequencies. This might suggest that coverage mapping may be sensitive to flight altitude in low-lying terrain, away from strong cellular readings.

The second location, *Urban*, was in a more urban area with better cellular coverage. In this area, the altitude did not alter the signal strength of frequencies sensed by the UAS. The fourth location, the *Hilltop*, was located high on a hill approximately 400ft above the *Residential* area. At this location, altitudes over 160ft showed a drop in signal strength across most of the monitored LTE frequencies. One possible explanation is that the aerial vehicle may have difficulty detecting coverage at altitudes significantly higher than the provisioned coverage area.

In addition to examining frequency fluctuations, we examine the received signal strength by the cellular network provider. In Figure 3.9 we show the mean change

in signal strength across all frequencies with altitude for each network and location. The change in signal is network-dependent, and the difference between networks depends on location. A probable explanation for the observed difference is that the eNodeBs serving these networks are in disparate geographic locations with different signal propagation patterns.

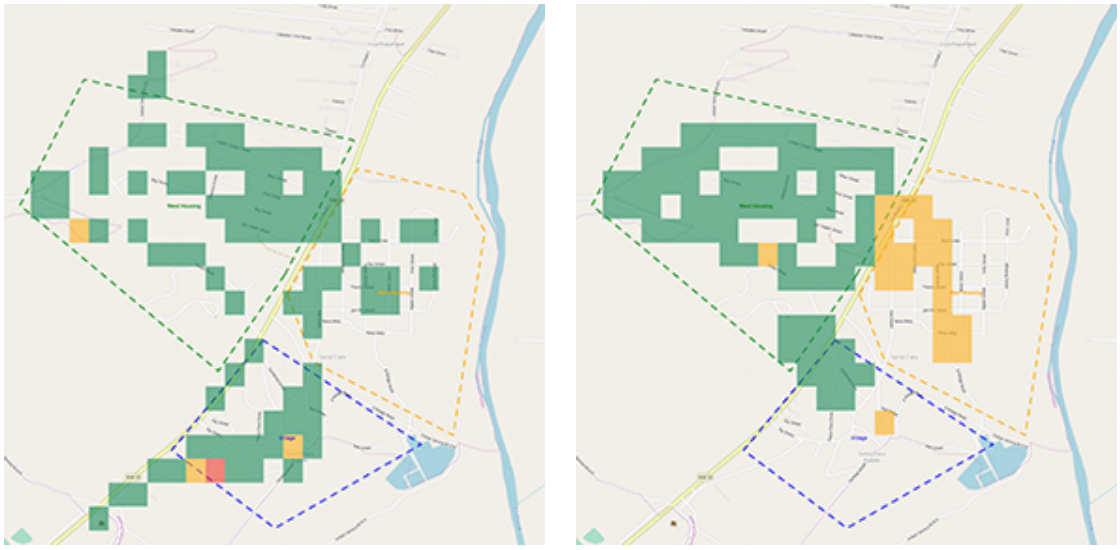
3.5 Discussion

While we observe clear relationships between sensing methods, the devices' relative signal strength values output are weakly correlated, particularly to the UEs, even after transforming the data to a typical reference frame. We believe the problem stems from the difficulty in aligning the various methodologies for comparison. Because we could not capture the frequencies on which the UEs operated, we compared the frequencies with the highest signal strength for a given method. This may not always match the actual frequency used by the UE. Additionally, the wardriving readings from the RTL-SDR, USRP, and SA are more difficult to collect due to the labor involved. As a result, there are fewer points of geographic overlap than for the UE and UAS measurements.

By categorizing individual signal strengths by quality, mirroring the "bars" of signal strength that a user's device might report, we could accurately match these categorical measurements across measurement methods. As the most versatile collection method, the UAS predicted quality within one gradation over 98% of the time. This aerial signal sensing method demonstrates promise as an effective system for wide-scale cellular coverage mapping.

We generated a coverage map for each method and provider based on our experimental data. Figure 3.10 shows a portion of the map for Verizon. The readings from

the UE are shown in Figure 3.10(a), and those taken from the UAS on the same day are shown in Figure 3.10(b). Colors and values correspond to Table 3.2, with high RSRP in green and low in red.



a. Map from UE readings b. Map from UAS readings

Figure 3.10: Cellular coverage map generation.

Next, we evaluated how the design of our aerial data collection impacted the accuracy and precision of UAS signal quality assessment. We considered: 1) how a sample taken at a particular time compares to the overall LTE channel quality during a 24-hour period; and 2) how different UAS altitudes impact this characterization.

Sensing over Time:

Consider the longitudinal analysis of the stationary sensing equipment from Section 3.4.2. For most frequencies, a single flight *is* sufficient; most readings fell within 3 dBm of each other. However, specific frequencies may be less stable. For example, two channels within the two-day deployment showed values varying by up to 7 dBm, which is wide enough to bump a reading by two signal quality levels (e.g., Good to Fair, or even Poor).

As observed in Figure 3.6, RTL-SDR signal strength measurements fluctuate between readings. The average of multiple readings provides a more stable description of signal quality in a geographic region. Because the UAS would need to take dozens of samples from a geographic region, the flight pattern and maximum flight speed would vary with the desired granularity of the measurements. The UAS must fly at low speeds to achieve high granularity (e.g., building-level accuracy) and at higher speeds to achieve greater coverage but lower granularity (e.g., neighborhood level). Alternatively, the UAS can fly at higher speeds following a flight plan that conducts repeated measurements at the same location (e.g., through flight loops).

RF signals sometimes fluctuate based on moisture and other weather conditions. In this study, we did not capture the sensitivity of RTL-SDR signal strength measurements to weather fluctuations and seasonal changes. The stationary sensing equipment was deployed on clear days with no rain, a daily temperature high of ≈ 72 and a low of ≈ 52. In future work, examining how signal readings from an RTL-SDR vary for much more extended periods would be informative. Such an assessment could reveal whether the RTL-SDR equipped UAS requires calibration depending on current weather. This study would also help understand how cellular network quality measurements from our system may fluctuate during or after a natural disaster.

Choice of Altitude:

To measure how altitude affects signal quality, we look to the analysis in Section 3.4.3. The interaction between altitude and signal strength reception by the UAS is complex. The local geographic topography seems to be the dominant factor in received signal strength. When flying in low valleys, an increase in altitude corresponded to an increase in mean received signal strength. Yet, signal strength declined when ascending from a hilltop above the residential area. Because network

providers' orientation of cellular tower antennas is provisioned to optimize coverage at elevations of residences and businesses [60], aerial collection at altitudes (in our case, approximately five hundred feet above the residential height) may see degradation in signal strength.

The effect of altitude on signal quality has implications for evaluating LTE coverage and availability for the occupants of high-rise buildings. We could use aerial systems in dense city centers to map signal strength in three dimensions. Conventional measurement methodologies would not account for such a measure of signal quality across floors in skyscrapers.

3.6 Conclusion

We have shown that a UAS-mounted RTL-SDR can provide a granular reflection of LTE signal strength. Our low-cost solution enables accurate coverage mapping and quality assessment in regions typically neglected by other assessment forms. Moreover, our system achieves this without requiring expensive specialized equipment, extensive time commitments, or significant manpower. This work aims to pave the way for future solutions that more accurately represent cellular coverage, particularly in those regions that are likely under-served. With that goal in mind, we build on this work to evaluate mobile broadband performance from ground-truth active measurements. We present our analyses and observations in the next chapter.

Chapter 4

Coverage is Not Binary: Quantifying Mobile Broadband Quality in Urban, Rural, and Tribal Contexts

4.1 Introduction

LTE plays an increasingly critical role in providing pervasive Internet access. A 2019 Pew Research study reports that roughly one in five US adults is "smartphone-dependent," meaning they solely rely on mobile broadband for Internet access at home [83]. Individuals living in rural and tribal regions are particularly likely to rely on mobile broadband for Internet access [45]. As growing numbers of people depend on LTE networks as their primary means for accessing healthcare, financial, and educational services, it has become *critical to evaluate how well these networks service user applications*. Due to the COVID-19 pandemic, the urgency of assessing the quality of experience for applications delivered over mobile broadband has skyrocketed as stay-at-home orders, and rapid movement to online schooling increase the demand

for applications that are known to be sensitive to network quality, such as video streaming and interactive video chat [5]. As a result, communities without access to usable, high-speed broadband, such as many rural and tribal regions, are severely disadvantaged [26].

There is a need for targeted measurement campaigns that represent performance within challenged networks [84]; the Federal Communications Commission recently encouraged researchers to undertake campaigns to study and report the state of rural networks [85]. Our goal is to understand the mobile quality of service (QoS) and quality of experience (QoE) performance profiles for common and increasingly essential applications such as video streaming and Web browsing, in a variety of network conditions. To do so, we undertook an extensive measurement campaign to collect 16 datasets of network traces in the Southwestern US for four major telecom operators: AT&T, Sprint, T-Mobile and Verizon, gathering over 30 million LTE packets. An overview of the datasets is provided in Chapter 2 (§2.1). We collected measurements of LTE networks in tribal, rural, and urban communities to understand geographic performance discrepancies. While we anticipate that network performance in tribal and rural areas will differ from that in urban areas because rural and tribal networks are often under-provisioned [6, 84], our study aims to quantify the severity of performance degradation in under-provisioned networks. Service quality is not a binary label, just like cellular coverage; for instance, application performance is subject to network conditions. Our goal to quantify network performance stems from the need to indicate the behavior of different applications accurately and not simply label a region as "covered" or "not covered."

Our tribal and rural measurements were conducted in New Mexico. New Mexico is one of the least densely populated states in the US, and 10% of its land area belongs to one of the 23 sovereign tribes with territories in the state [86]. In the

rural regions, there is a high concentration of smartphone-dependent residents [87]. In addition to the tribal and rural contexts, we collect network traces from crowded events in urban locations in California, during which atypically high volumes of network utilization cause congestion. For comparison, we also collect traces from the exact urban locations during typical operating conditions as a baseline. Our datasets have broad spatial and temporal variability but can be classified into three categories: under-provisioned (rural and tribal), congested (congested urban), and well-provisioned (baseline urban).

While Web browsing is a critical component of daily Internet access, streaming video currently accounts for 65% of all downstream mobile traffic worldwide [88] (for instance, in the US, more than 80% of the population possess some form of video streaming subscriptions [89]). Therefore, we focus our analysis on understanding web browsing and streaming video QoE for these regions. At each location, we collect extensive QoS and QoE measurements. Based on our analysis, we illustrate critical performance differences between the three location categories. Our key contributions and findings include:

- Collection of 16 network performance datasets from 12 locations across the Southwestern US, representative of three network conditions: under-provisioned (rural and tribal), congested urban, and well-provisioned urban;
- Characterization of LTE traffic across all locations and network conditions in the datasets, through analysis of four QoS and six QoE metrics;
- Analysis of QoS and QoE data reveals that rural and tribal LTE networks consistently perform worse than the studied urban baseline deployments and are typically comparable to or worse than congested urban networks.

Type	Metric	Test Interval	# of Datapoints	Tools
QoS	RSRP	1 second	2160	Network Monitor
	Throughput	1 second	2160	iPerf
	Latency	1 second	960	HPing3
	Packet loss	N/A	N/A	tshark
QoE	Start-up delay	1 second	2160	Selenium, iframe API
	Video quality	1 second	2160	Selenium, iframe API
	Resolution switches	1 second	2160	Selenium, iframe API
	Rebuffering percentage	1 second	2160	Selenium, iframe API
	Buffer size	1 second	2160	Selenium, iframe API
	Page load time	N/A	300	Selenium, Chromium

Table 4.1: Overview of QoS and QoE metrics at each location, aggregated across available providers.

4.2 Evaluation Metrics

Typically, a binary representation of cellular coverage (i.e., is an area covered or not) is used to characterize the state of Internet connectivity over LTE networks. However, from our own experiences, as well as that of others [10], such a simplistic characterization of networked services over LTE networks is insufficient. This becomes increasingly true in rural regions, as base station coverage areas are more significant and weaker signals are more commonly experienced. Even in well-covered urban areas, performance can suffer during times of atypically high usage, i.e., flash crowds due to a heavily attended community event [90, 91]. As application requirements place more load on the network, it becomes critical to determine whether coverage exists in a region and whether that coverage is of high enough quality to support the types of applications wanted or needed by the local users. As we have seen with recent shelter-in-place orders due to COVID-19, residents of regions with sub-standard Internet access are at risk of being left behind educationally, economically, and medically [25, 26]. To evaluate network quality, we turn to QoS and QoE metrics and use these metrics to analyze the ability of the networks to support the most accessed

applications: Web browsing and video streaming traffic, which are applications of high, and still growing usage. In this section, we describe the QoS and QoE metrics collected for this measurement study, summarized in Table 10.1.

4.2.1 Quality of Service Metrics

Different applications have different network requirements, and QoS metrics capture the state of network performance. For instance, delay-sensitive Internet traffic, such as live streaming video and multimedia teleconferencing, requires low end-to-end delay to maintain interactivity; an application such as on-demand gaming is dependent on both end-to-end delay and achieved throughput. Barriers to attaining QoS in LTE networks include poor coverage quality and high network utilization. We measure four metrics to determine QoS.

Reference Signal Received Power (RSRP): RSRP is defined as the linear average over the power contributions (in Watts) of the resource elements that carry cell-specific reference signals within the measurement frequency bandwidth [92]. Although many key performance indicators (KPIs) are related to received signal strength, we focus specifically on RSRP, as defined by 3GPP [93]. [94] demonstrates that RSRP significantly impacts the mean opinion score (MOS) of video streaming; MOS varies greatly at RSRP values between -84dBm and -102dBm and declines rapidly below -104dBm. RSRP is used for a variety of LTE operations (e.g., cell selection, handover decisions [69], network quality assessment, etc.) and, as illustrated by [95], is widely accessible through mobile operating systems. Typically, RSRP is reported in dBm by the user equipment (UE) as the average power over several narrow-band control channels. We record instantaneous RSRP readings from the user equipment every second through the Network Monitor application [79].

Throughput: Our network monitoring suite automates the collection of throughput measurements by fetching a pre-specified 500 MB file from an AWS instance hosted in Virginia. For uniformity, we use the same example for all measurement tests. To calculate the throughput, the client initiates iPerf threads over TCP to download the file. The large file size allows the data traffic to fill the pipe and minimize the effect of a slow start. We log the packet traces at the client during the iPerf tests to sample throughput over one-second intervals for each location.

Latency: We measure round-trip times through pings, initiated by Hping3 [96], to the same Virginia-based AWS server. We configure Hping3 to use TCP packets instead of ICMP because ICMP packets were occasionally dropped at the server. The latency test runs for 120 seconds with one-second intervals between each ping. We measure latency twice during each measurement session: once before the video stream and throughput tests (described above) and once immediately after. Hence we collect 240 latency datapoints per operator, for 960 total at every location. Low round-trip times tend to be indicative of a better user experience for delay-sensitive applications.

Packet Loss: Packet loss in cellular networks can occur due to network congestion and transmission errors [97]. We analyze the synchronous packet traces from both the client and the server during throughput tests to compute packet loss using `tshark CLI`.

4.2.2 Quality of Experience Metrics

To measure QoE, we focus on streaming video and Web browsing, currently the most heavily used QoE-centric applications in mobile networks [98]. Internet video streaming services typically use Dynamic Adaptive Streaming over HTTP

(DASH) [99] to deliver a video stream. DASH divides each video into time intervals known as segments or chunks, encoded at multiple bitrates and resolutions. We measure various metrics associated with video streaming quality, as described below.

To assess web browsing quality, we measure the page load times of some of the most frequently accessed Web pages. Numerous studies and media articles report its importance for user experience [100, 101], and consequently to business revenue. The QoE metrics we measure are summarized in Table 10.1 and are described below. Our approach for measuring the majority of these metrics is described in section 10.2.2.

Start-up Delay: Start-up delay is the time elapsed from when the player initiates a connection to a video server to when the application starts rendering video frames. This delay usually corresponds to how quickly the HTTP adaptive streaming client can fill the threshold buffer required for playback.

Video Quality: Video quality is the number of pixels in each dimension of a video frame [102]. Video quality, or resolution, is an essential component of QoE; a higher resolution results in a better visual experience up to a point.

Resolution Switching: Frequent changes in video resolution can result in user frustration, mainly when the video quality is downgraded [98]. We compute the number of samples that had a different resolution from the prior sample in our video streaming sessions as a percentage of the total number of samples collected during the video session. Since resolution switches occur in-between video *chunks* that are typically 4–5 seconds long [102], our analysis at one-second granularity is a lower bound estimate, if not better. Both the magnitude (difference in pre-and post-switch resolution) and the frequency of video resolution switches affect the quality of experience [98].

Rebuffering Percentage: A rebuffering occurs when the application buffer waits to accumulate enough content to resume playback. Poor link quality and congestion can

increase video rebuffering events because they cause delays in packet delivery [103]. When rebuffering occurs, the user notices interrupted video playback, commonly referred to as *stalling*. Rebuffering events have a crucial influence on user satisfaction and significantly impact video abandonment [104]. We represent the rebuffering percentage as the amount of time the video stalls during the playback expressed as a percentage of total playback time.

Buffer Size: The streaming client employs a playout buffer or client buffer, whose maximum value is the buffer capacity (in seconds) to temporarily store chunks to absorb network delay variations (i.e., jitter). To ensure smooth playback and adequate buffer level, the client requests a video clip chunk by chunk (in seconds) using HTTP GET requests and dynamically determines the resolution of the next chunk based on network conditions and buffer status. When the buffer level is below a minimum threshold, the client requests chunks as fast as the network can deliver to increase the buffer level. The playback stalls when the buffer is empty before the end of the playback is reached.

Page Load Time: To compute load times, the fetching of Web pages is automated using Selenium [105]. We use the Tranco Top 25 list [36]. For evaluation, we log the navigation timings of a Web page starting from *navigationStart* through the *loadEventEnd* event [106]. These event timings help in a fine-grained analysis of page load times. We download each webpage three times and average the results. The browser cache is automatically wiped out after each Web page load to reflect the true load time for the next iteration.

4.3 Methodology and Datasets

It is our goal for our measurements to represent a range of network deployments that vary both by signal quality and offered load. We focus on the networks of the four major U.S. providers: AT&T, Sprint, T-Mobile, and Verizon. In this section, we first describe our custom measurement suite and our measurement methodology. We then describe the details of our collected datasets.

Location	Date (2019)	# LTE Packets	Type	Carriers*
New Mexico				
Tribal_1	May 28	3.18 Million	Tribal, Rural	V,A,T,S
Tribal_2	May 29	1.38 Million	Tribal, Rural	V,T
Tribal_3	May 28	2.03 Million	Tribal, Rural	V,A,T,S
Tribal_4	May 30	2.16 Million	Tribal, Rural	V,A,T,S
Tribal_5	May 30	2.27 Million	Tribal, Rural	V,A,T,S
Tribal_6	May 31	2.33 Million	Tribal, Rural	V,A,T,S
Rural_1	May 31	1.26 Million	Non-Tribal, Rural	V,T
Rural_2	Jun 01	2.83 Million	Non-Tribal, Rural	V,A,T,S
San Diego, CA				
Urban_1_Cong	Sep 22	2.25 Million	Urban, Congested	V,A,T,S
Urban_1_Base	Sep 28	1.92 Million	Urban, Baseline	V,A,T,S
Urban_2_Cong	Sep 29	2.51 Million	Urban, Congested	V,A,T,S
Urban_2_Base	Sep 30	1.97 Million	Urban, Baseline	V,A,T,S
Urban_3_Cong	Sep 21	2.65 Million	Urban, Congested	V,A,T,S
Urban_3_Base	Sep 30	2.13 Million	Urban, Baseline	V,A,T,S
San Francisco, CA				
Urban_4_Cong	Sep 25	2.18 Million	Urban, Congested	V,A,T,S
Urban_4_Base	Sep 26	2.08 Million	Urban, Baseline	V,A,T,S

Table 4.2: Summary of Datasets
*This column lists the mobile carriers present in each data set (some areas had no coverage for particular network operators). V: Verizon, A:AT&T, T:T-Mobile, S: Sprint.

4.3.1 Measurement Suite

We collect measurements from which we can derive QoS and QoE in various locations in which we expect varying LTE performance. The capture of disparate

performance will give us a broad picture of QoS and QoE and the opportunity to study performance in different population densities (urban vs. rural vs. tribal reservations) and usage scenarios (congested vs. typical usage). To collect these measurements, we use a custom-built measurement suite that captures both network- and application-level metrics. For video, application-level metrics are measured by streaming YouTube videos; we choose YouTube because it has an extensive mobile reach of 88%, more than 3.5× that of its next best competitor [107]. For Web browsing, we utilize the Tranco ranking list as it addresses the stability and responsiveness issues faced by other Website ranking services [36].

We run our measurement suite on Lenovo ThinkPad W550s laptops, each tethered to its own Motorola G7 Power (Android 9) via USB to measure cellular performance. The cellular plans on all our cellular user equipment (UE) have unlimited hot-spot data enabled to effectively achieve the same level of performance as we would on the mobile device. We run our measurement suite on laptops tethered to phones as this configuration gives us the same application performance while facilitating ease of programming and data extraction and achieving a higher level of unification for various application-level measurements. We record instantaneous RSRP readings from the UEs every second through the Network Monitor application [79].

To collect video QoE metrics, we run a 3-minute clip of a Looney Tunes video three times across each of the four LTE providers at each location; we exclude from our results the sessions that experienced playback errors during execution. We chose this particular video due to its mix of high and low-action scenes, which results in variable bitrates for different segments in the video (typically, high-action scenes have higher bitrate than low-action scenes). To infer video QoE, we collect the input features (RSRP and throughput) synchronously on a separate device so as not to bias the video streaming measurements. After testing multiple playback durations, we

observed that a 3-minute window was adequate for the playback to reach steady-state while long enough to capture rebuffering and resolution switches. To execute this experiment, we first automate the loading and playback of the YouTube video on the Chrome browser using Selenium [**?**]. The video resolution is set to auto. Then we use YouTube's iframe API [108] to capture playback events reported by the video player. The API outputs a set of values that indicate player state (not started, paused, playing, completed, buffering) using the getPlayerState() function. The API also provides functions for accessing information about playtime and the remaining buffer size. Similarly, we employ Selenium to automate Web page loading on the Chromium browser to capture page load time measurements.

4.3.2 Description of Datasets

We collected 16 datasets from 12 locations across the Southwestern US Eight of those were collected from rural locations with sparse cellular deployment. Six of the eight locations were under the jurisdiction of American Indian Tribal sovereignty, while two were from non-tribal rural regions. In the remaining text, we sometimes use the word "rural" to refer to tribal and non-tribal rural areas. These eight collection points spanned 21 square miles in New Mexico and were collected for five days. In addition, we collect another eight datasets from four urban locations in California. For each urban location, we collect two datasets: one during a large event or gathering, in which we expect cellular network congestion to occur (these datasets are marked with **_Cong**) [90, 109]; and a second during typical operating conditions. We call the latter dataset the baseline for that location (marked with **_Base**). Hence, our traces are broadly classified into three categories: rural and tribal, congested urban, and baseline urban. The details of each dataset are summarized in Table 10.2. In

each location, we concurrently collected the complete traces on four major U.S. carriers (AT&T, Sprint, T-Mobile, and Verizon) using four separate, equivalent UEs, as described in §4.3.1. The designation of each location as tribal, rural, or urban ("type" column in Table 10.2) is based on information gathered from the Census Bureau [38]. We collect and analyze over 32.7 million LTE packets through these measurement campaigns. A detailed overview of the datasets is provided in Chapter 2 (§2.1)(Santa Fe, New Mexico; Suburbs, San Diego; San Francisco).

4.4 Network Characterization

In this section, we analyze network performance characteristics in each measurement location, and by so doing, we attempt to determine whether any generalizations based on network offered load exist. Note that one or more operators did not provide LTE coverage in specific locations, as indicated in Table 10.2. Our assessment reveals several differences in network performance across region types and network conditions (congested/uncongested).

4.4.1 Quality of Service Analysis

We evaluate the relationship between spatially and temporally varying network conditions through four QoS metrics: RSRP, throughput, latency, and packet loss, as described in §4.2.1. We analyze the mean and median values and present median results aggregated across all telecom providers at each location. In addition, wherever applicable, we report performance deviations that fall beyond one standard deviation (68% confidence interval) because they may skew the distribution. Stated otherwise, we separately report the performance of each telecom operator if

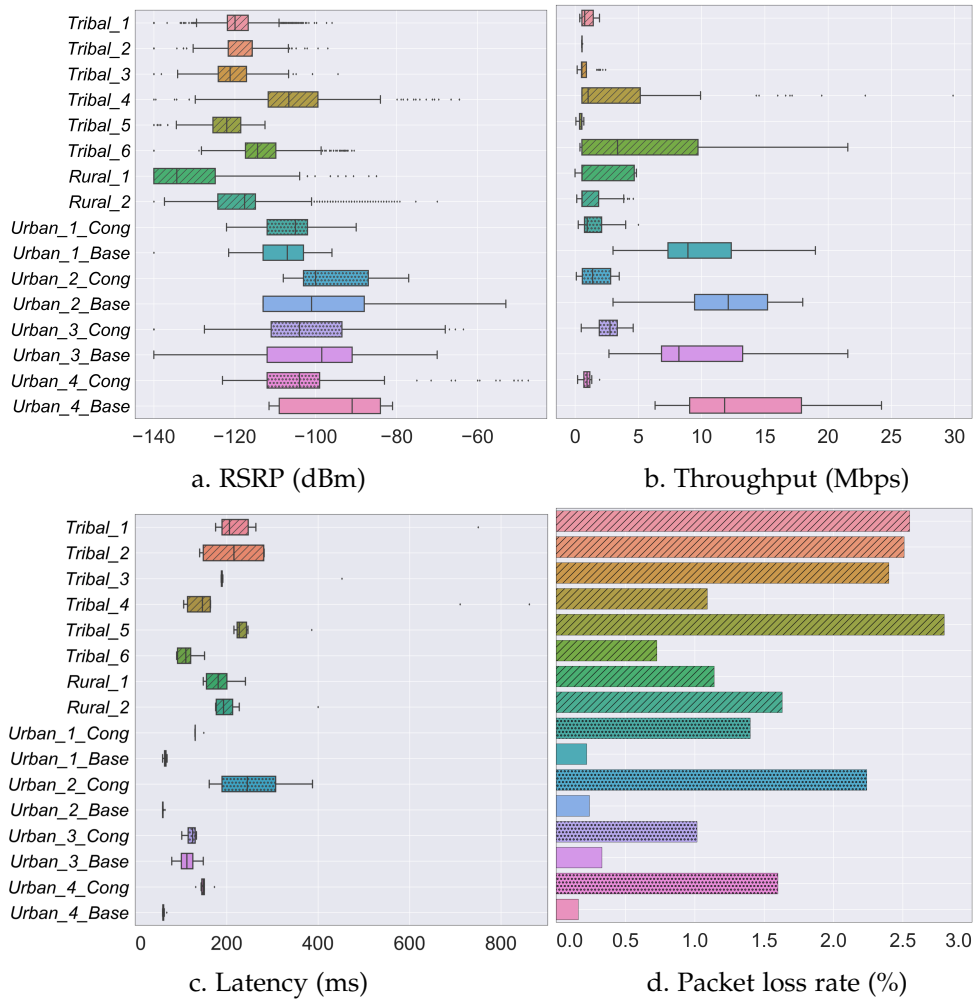


Figure 4.1: Distribution of QoS metrics across different network conditions.

$x \leq \mu - \sigma$ or $x \geq \mu + \sigma$, where x is the median performance value for the outlier operator, μ and σ are the mean and standard deviation of the entire distribution. In the boxplots, the right and left edges of the box represent the first quartile (25^{th} percentile) and third quartile (75^{th} percentile), respectively, with the median line drawn within the box. The whiskers capture 5^{th} to 95^{th} percentile values.

RSRP: We observe a wide range of RSRP values on all levels: between datasets, within datasets, and between dataset types, as shown in Figure 4.1(a). The median RSRP value across all rural and tribal locations is -118dBm. We observe that rural

and tribal areas report 15.06% and 19.55% lower RSRP values than urban congested and baseline urban locations, respectively.[1] This result is consistent with reports of limited LTE coverage in rural and tribal locations; these regions frequently have significantly sparser LTE base station deployment and hence larger coverage areas that lead to regions with lower signal quality. One notable exception is Tribal_4, where the reported values are, on average, 12dBm higher than elsewhere in the rural region. This is likely due to the relatively denser deployment of base stations in the area; our wireless gear was physically closer to the eNodeBs (LTE base stations) that served the region (verified through CellMapper [50]).

Throughput: Figure 4.1(b) compares throughput across all locations, averaged across all providers in each location. We observe high variability in throughput distributions, ranging from less than 500 Kbps to about 30 Mbps. Congested urban measurements report a median throughput of 1.51 Mbps, while rural and tribal regions report 35% less median throughput, at 0.98 Mbps. Uncongested urban locations have by far the best average performance.

We observe a few outliers in our analysis: AT&T reports 30× and 26× the median throughput values in Tribal_4 and Tribal_6, faring considerably better than its competitors. In addition, Sprint performs 8× better than the region's median. In Tribal_4, if we exclude the outlier (AT&T), the median value for the other operators is 0.56 Mbps, which is 41% less than the worst-performing congested dataset, Urban_4_Cong (median: 0.95 Mbps). This is unexpected since: (1) Tribal_4 has denser coverage, and (2) our measurement setup was close to all the connected eNodeBs

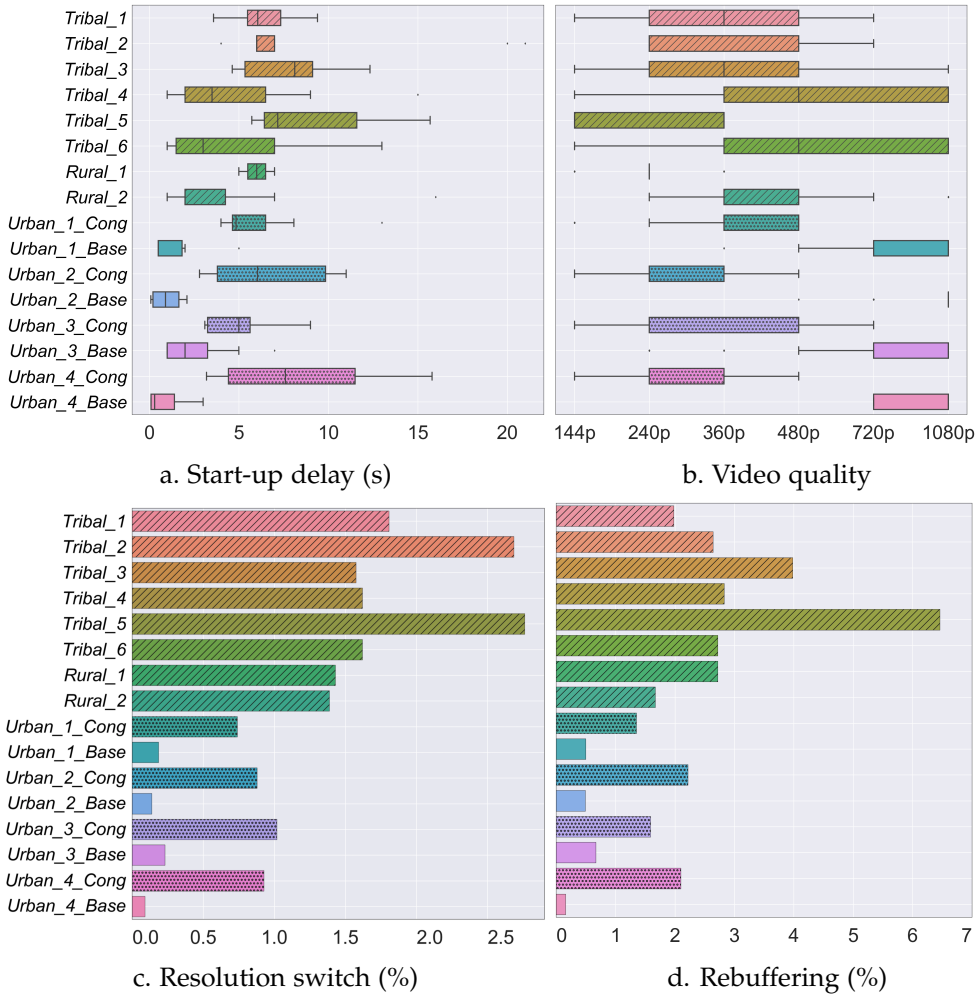
[1]The slight difference in RSRP median value between the congested urban and baseline results may stem from multiple causes: a difference in weather conditions on measurement days, a change in transmission power of the eNodeBs due to utilization, and the fact that the Urban_4_Base collection location was approximately 30m away from the Urban_4_Cong location due to the closure of the original collection venue.

(this resulted in 12dBm higher RSRP than other rural locations). Similarly, if we exclude the outliers in Tribal_6 (AT&T and Sprint), the median throughput is 43% worse than Urban_4_Cong.

Our results demonstrate that, on average, the LTE networks we measured in rural regions perform worse than congested urban networks. This performance difference, and the absolute performance values, likely indicate the difference between having a stable teleconferencing session and an unusable service (e.g., Zoom recommends a minimum downstream bandwidth of 1.5 Mbps [110]). Urban baseline locations register a median throughput of 10.92 Mbps. In comparison, congested locations report 7× less median throughput, while rural and tribal regions register 11× lower throughput than baseline urban locations.

Latency: Figure 4.1(c) shows the average latency, measured as round-trip time (RTT), across all measurements in each location. Urban baseline datasets reveal a median latency of 64 ms. At the same time, in the congested networks, the average RTT doubles to a median value of 140 ms, verifying our expectation of network congestion. Rural regions report a median latency of 193 ms, which translates to a 38% and 202% increase in round trip times compared to congested and urban baseline datasets, respectively. Notably, Tribal_4 has an average latency of 147 ms despite the proximity of our measurement setup to the LTE base stations and a location geographically closer to our ping server in Virginia than the locations in California. Reasons for this extra latency are varied and may include a less direct and slower path out of this region to a major Internet backbone. Overall, Tribal_2 and Urban_2_Cong exhibit the widest variability in latency measurements.

Packet Loss: The average loss rate across providers is reported in Figure 4.1(d). While we observe variability, a birds-eye view demonstrates that the baseline ur-



a. Start-up delay (s)

b. Video quality

c. Resolution switch (%)

d. Rebuffering (%)

Figure 4.2: Distribution of QoE metrics across different network conditions.

ban locations benefit from marginal loss rates (median is 0.23%). We observe more than 6× higher median loss rates in the congested urban datasets, indicating heavy congestion. As a group, the congested urban locations experience the second-best performance (median loss rate is 1.56%), and rural networks experience the highest average loss rates. We observe an exceptionally high loss in Tribal_1, Tribal_2, Tribal_3, and Tribal_5 (median of 2.53% across all four locations). A performance outlier is Tribal_6; despite its comparable rural and tribal location, Tribal_6 reports a much lower loss rate of 0.72%.

Takeaway: Our analysis of QoS metrics reveals the wide gap in LTE performance across different regions and network conditions. Our results illustrate that the rural and tribal regions we study experience the poorest mobile broadband performance. This performance is typically even worse than heavily congested urban networks. This poor performance is consistent with prior findings [87, 111]. The chronic under-provisioning of LTE networks in rural and tribal regions, due to both sparser deployment and some combination of less efficient and lower speed backhaul, implies mobile broadband in these regions often cannot meet the minimum recommended QoS required for applications such as video streaming and video chat. While this poor Internet usability has a negative impact on residents, this impact has been grossly amplified during the shelter-in-place orders of the COVID-19 pandemic, when schooling, work, telemedicine, and other critical activities have been moved online [25, 26]. Our measurements indicate that in many of these rural locations, despite the presence of mobile broadband, the quality of those networks is often too poor to support these now-essential video-based applications.

4.4.2 Video QoE Analysis

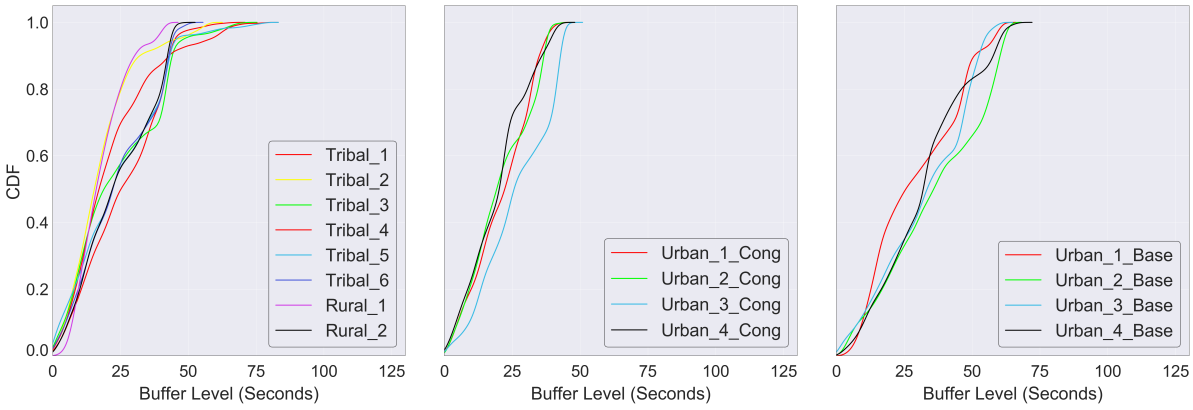
Next, we characterize key video QoE indicators in different network conditions to reveal application-level performance differentials of video streaming. Similar to §4.4.1, we report median values across all telecom providers unless there are samples that lie outside of one standard deviation (μ).

Start-up Delay: Figure 4.2(a) plots the start-up delay. The median start-up delay in rural and tribal regions is 6.52s, while congested urban locations report 5.29s. Urban baselines have the lowest reported delay at 0.7s (median). We also note that the rural and tribal locations have far higher variability than congested datasets. For instance,

the median range of start-up delay (i.e., the difference between the max and the min values in a distribution) is 12.5s in these areas as opposed to 8.6s in congested networks. This behavior can be attributed to under-provisioned LTE networks in rural and tribal regions that are sensitive to user demand, even during normal operating hours, resulting in large fluctuations. This result is consistent with observations in Figure 4.1(b); lower throughput coupled with higher packet loss would likely result in the inconsistencies in the download time of the initial video segments. Baseline urban offers the least variability of 3.7s. In Figure 4.1(b), we saw that the AT&T network achieves higher throughput at Tribal_4 and Tribal_6. Consequently, we observe that video sessions in both Tribal_4 and Tribal_6 on the AT&T network have 3$\times$ lower start-up delay than the other providers. We note that the start-up delay does not convey any information about playback video resolution.

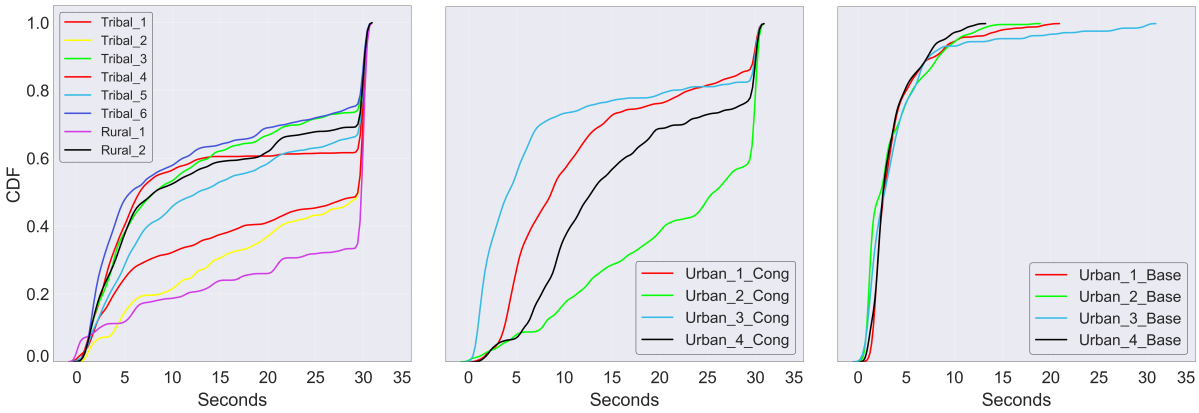
Video Quality: Figure 4.2(b) depicts the playback resolution of the YouTube video, sampled at one-second granularity. We ensure that the video resolution is set to auto during our measurements. As a result, playback resolution and resolution switches are a direct result of network conditions and changes in congestion levels. While all of the urban baseline locations indicate near full-HD (1080p) rendering of the video, congested locations have a median resolution of 240p. One possible explanation could be telecoms' throttling of video quality as part of their congestion mitigation schemes. Rural measurements show marginally better performance with a median resolution of 360p but exhibit wider variability, ranging between 144p to 1080p. Video sessions with 1080p in rural regions are associated with the AT&T network, consistent with the results from Figure 4.1(b), where AT&T records distinctively higher records throughput values.

Resolution Switching: Video resolution variation is synonymous with quality switches,



a. Rural and tribal b. Congested urban c. Baseline urban

Figure 4.3: Cumulative distribution of buffer size across different network conditions.



a. Rural and tribal b. Congested urban c. Baseline urban

Figure 4.4: Page load times of Tranco top 25 websites.

which is often perceived as a QoE performance degradation [112]. Figure 4.2(c) represents the number of samples with a different resolution than the previous sample as a percentage of total collected samples during the video session. We observe a median value of 1.64% in rural regions, as opposed to 0.92% in congested urban datasets. This value is nearly 6× smaller for baseline urban datasets, with a median value of 0.16%, compared to urban congested measurements. Frequent resolution switches typically lead to lower user engagement [112]. This implies that the high

percentage of switches in rural and tribal regions could lead to difficulty in user engagement with video streaming services, as needed during remote learning, work-at-home video conferences, and other vital applications.

Rebuffering: This metric represents the time the video stalls or rebuffers during the playback expressed as a percentage of total playback time, shown in Figure 4.2(d). There is a higher median rebuffering ratio of 2.68% in rural and tribal regions, followed by congested urban regions at 1.85%. Baseline urban measurements report a more than 5× and 3× lower rebuffering ratio than rural/tribal and congested urban, respectively.

Buffer Size: Figure 4.3 shows the buffer size distribution captured during YouTube streaming sessions. For ease of comprehension, we separate the result graphs into three categories. Here, a greater amount of buffered content is better, allowing the application to smooth performance despite varying jitter. Baseline measurements report higher buffer levels with a median value of 44.3s (Figure 4.3(c)) while congested datasets report 27.7s (Figure 4.3(b)), a 34% decrease from baseline measurements. Rural regions have the lowest median buffer at 20.2s (Figure 4.3(a)), which is a reduction of 52% and 27% from baseline urban and congested urban measurements, respectively.

Takeaway: Our analysis of QoE metrics indicates that user experience suffers due to under-provisioned LTE networks in rural and tribal regions. The results reinforce our findings in §4.4.1 that LTE networks in these regions are likely to fail to provide a quality, or even usable, experience for video streaming. Unsurprisingly, rural networks underperform in most cases compared to congested LTE networks in urban regions, implying that *the worst-case experience in an urban network is likely still better than the average case experience in a rural or tribal region*. However, the extent of perfor-

mance degradation in rural and tribal areas compared to other network conditions is remarkable and noteworthy.

4.4.3 Page Load Time

Locations	PLT Timeout	Locations	PLT Timeout	Locations	PLT Timeout
Tribal_ 1	16.9%	Rural_1	66.1%	Urban_3_Cong	17.3%
Tribal_2	51.6%	Rural_2	27.3%	Urban_3_Base	0.88%
Tribal_3	26.4%	Urban_1_Cong	17.6%	Urban_4_Cong	24.0%
Tribal_4	46.7%	Urban_1_Base	0.0%	Urban_4_Base	0.0%
Tribal_5	30.1%	Urban_2_Cong	41.8%	–	–
Tribal_6	24.6%	Urban_2_Base	0.0%	–	–

Table 4.3: Webpage Load Timeouts.

Web performance has long been crucial to the Internet ecosystem since a significant fraction of Internet content is consumed as Web pages. The end-user quality perception in interactive data services is dominated by Web page loading times; the longer they wait, the lower the user satisfaction [113]. Studies have shown that *perceived* time for users accessing the Web can be exceedingly magnified for actual chronological time, thus degrading the *perceived* performance even further [114, 115]. Page load times are depicted in Figure 4.4. We bin the results into similar categories as in Figure 4.3. From our evaluation, we learn that rural and tribal locations (shown in Figure 4.4(a)) produce the slowest load times with a median value of 9.75 seconds. This is 74% slower than the congested dataset (median value is 5.6 seconds) and 2.7$\times$ lower than baseline urban measurements (median value is 2.67 seconds). Tribal_2 performs the worst in tribal and rural locations with a median load time of 13.08 seconds, while Urban_2_Cong is the most under-performing dataset in congested urban at 7.28 seconds. In baseline urban, all locations exhibit similar load times within a margin of $\pm$1 second difference. Our examination reveals a considerable fraction of

Web pages fail to load within the timeout period of 30 seconds (shown in Table 4.3) in rural, tribal, and congested urban regions. We observe a median timeout value of 28.7% across tribal and rural areas, which is 38.7% higher than that reported in congested urban (median value is 20.8%). Rural_1 logs the highest timeout percentage with over 66% of Web pages failing to load; Urban_2_Cong reports the highest (41.8%) in the congested dataset. Urban baseline locations have faster load times and little to no timeouts for Web pages. Similar to section 4.4.2, we observe that the Web browsing experience suffers more in rural and tribal regions than in urban regions (with or without atypical network utilization).

4.5 Related Work

Manual measurements are a common approach to calculating cellular coverage [116]. This includes methods such as war-driving [51], war-walking [117], and aerial systems [35], which usually require high operational expenditure. Mobile analytics companies [33, 118] contribute to measurement collection by crowd-sourcing measurements directly from end-user devices via standalone mobile apps [33, 119] or measurement SDKs [120, 121, 119] integrated into partner apps. However, these are limited in scope because crowd-sourced measurements do not have spatial uniformity. As a result, some of the desired measurement locations may not exist in these databases (possibly due to a lack of adoption of the application or SDK by the local community). Further, outsourced databases typically carry a hefty licensing fee or are otherwise restricted [122]. While several public datasets consist of Internet performance measurements (e.g. [123, 121]), there is a lack of datasets that represent the variability in mobile broadband performance due to sparse deployments or network congestion. Many mobile network datasets focus on coverage [33, 34]; the FCC an-

nually publishes its broadband report [6]. Unfortunately, these broadband reports are widely inaccurate [9]. Further, we have shown that using coverage maps alone is inadequate to infer actual usability. While several prior studies [124, 125] have focused on LTE performance analysis and traffic characterization, these studies do not compare performance across differing population densities and region types.

4.6 Conclusion

Online learning, work-at-home, telemedicine, and other applications that have already experienced regular usage have exploded in the post-COVID-19 world, transitioning from conveniences to critical everyday applications. Web browsing and video streaming are necessary components of these applications, so studying network performance for users in all regions is crucial. Through extensive measurements, we have revealed the sharp contrast in cellular performance between rural, tribal, and urban locations; for instance, video QoE is at least 10× worse, and Web browsing is more than 2× worse in the rural and tribal regions we studied than in urban locations with the typical cellular load. While prior work and past surveys have reached similar conclusions, our study demonstrates and quantifies the extent to which network performance lags in rural and tribal communities. This suggests that users in underserved regions are far more likely to drop out of virtual engagement such as online lectures and e-learning. User disengagement will, unfortunately, lead directly to a greater digital gap than exists today [25, 26]. Broadband deployments that address these access and coverage quality gaps are urgently needed. In the following chapter, we discuss the significant limitations of coverage maps that offer a binary assessment of service quality. We discuss our recommendations for improved cellular coverage measurements based on our observations.

Chapter 5

A Tale of Three Datasets: Towards Characterizing Mobile Broadband Access in the United States

5.1 Introduction

Affordable, quality Internet access is critical for full participation in the 21st-century economy, education system, and government [126]. Mobile broadband can be achieved through commercial Long-Term Evolution (LTE) cellular networks, which are a proven means of expanding this access [18], but are often concentrated in urban areas and leave economically marginalized and sparsely populated areas underserved [6]. The U.S. Federal Communications Commission (FCC) incentivizes LTE operators serving rural areas [29, 30] and maintains transparency by releasing maps from each operator showing geographic areas of coverage [46]. Recently third parties have challenged the veracity of these maps, claiming these maps over-represent true coverage, and thus may discourage much-needed investments.

Most of these claims, however, are either focused on limited areas where a few dedicated researchers can collect controlled coverage measurements (e.g., through wardriving) or are mainly qualitative in nature [31, 10, 32]. *As dependence on mobile broadband connectivity increases, especially in the face of the COVID-19 pandemic, mechanisms that quantitatively validate FCC coverage datasets at scale are becoming acutely necessary to evaluate and direct resources in Internet access deployment efforts [83, 127].* This is an issue of technology and technology policy, with equity and fairness implications for society.

An increasingly widespread approach to measuring coverage at scale is through crowdsourcing, wherein users of the LTE network contribute to coverage measurements. The FCC has recently advocated for crowdsourcing to validate coverage data reported by operators [128]. In this context, we take a data-driven, empirical approach to this work, comparing coverage from a representative crowdsourced dataset with the FCC data. More specifically, our analysis is guided by the following questions: (i) How consistent are existing LTE coverage datasets, ii) where and how do their coverage estimations differ, and what trends are present?

We specifically consider a crowdsourced coverage estimate from Skyhook, a commercial location service provider that uses various positioning tools to offer precise geolocation. We select Skyhook because it crowdsources cellular coverage measurements from end-user applications that subscribe to its location services. Such *incidental* crowdsourcing can potentially provide richer coverage data than a *voluntary* form of crowdsourcing where a user has to commit to contributing coverage data explicitly. We examine this by comparing the Skyhook measurements with those of OpenCellID, an open but voluntary crowdsourced dataset [129]. As will be shown in Section 5.3.1, we find that the density of the crowdsourced datasets varies significantly by the data collection methodology, especially in rural areas. In the regions

we studied, incidental crowdsourcing (Skyhook) gathered up to 11.1x more cell IDs than voluntary crowdsourcing (OpenCellID).

Using Skyhook as an extensive crowdsourced dataset, we quantify how widely and where the crowdsourced coverage data differs from the FCC data. We specifically focus on the state of New Mexico[1], selected for its mix of demographics, diverse geographic landscape, and our partnership with community stakeholders within the state. We compare coverage at the level of census blocks[2] which are further grouped into urban, rural, and tribal[3] categories. We find that the FCC and Skyhook LTE datasets disagree as great as 15% in rural census blocks, with the data from FCC claiming higher coverage than Skyhook. A significant concern in interpreting this comparison is accounting for coverage disagreement due to the lack of data points in the crowdsourced dataset. To confirm the availability of users to provide data points, we check for the presence of alternate cellular technologies (e.g., 2G or 3G) within these census blocks and observe a significant number (up to 9% in tribal, rural areas) where such alternates are present, providing evidence that users do visit those blocks but cannot access LTE. These results, similar to a recent study on fixed broadband [131], suggest a need for incorporating mechanisms to validate the operator-submitted data into the FCC's LTE access measurement methodology, especially in rural and tribal areas.

Finally, we compare both FCC and Skyhook coverage maps to our controlled coverage measurements collected from a northern section of New Mexico. Interestingly, we find that both FCC and Skyhook datasets report higher coverage relative to our

[1]Our methodology is not specific to New Mexico and can be easily extended to other regions in the US

[2]We use the FCC methodology wherein a census block is considered covered if the centroid is covered [130]

[3]Tribal areas that have consistently experienced the lowest broadband coverage rates in the United States for the past decade [6]

Data Set	Points of Collection	Format	Methodology
FCC	Polygon overlay	Shapefile	Operator-reported with Form 477
Skyhook	Cell signal point	CSV	Incidental crowdsourcing
Author controlled measurements	Cell signal point	CSV	Wardriving

Table 5.1: Summary of coverage data sets.

controlled measurements, with the former showing a higher degree (up to 26.7%) of over-reporting than the latter. Understanding the causes of these inconsistencies is crucial for effectively using crowdsourced data to measure LTE coverage, especially as crowdsourcing is increasingly viewed as preferable to provider reports. We conclude with recommendations for improving LTE coverage measurements, whose importance has only increased in the COVID-19 era of performing work and school from home.

5.2 Background and Datasets

In this section, we first provide an overview of the LTE network architecture. This is followed by a description of the LTE coverage datasets compared in our analysis. These datasets are summarized in Table 5.1. We also note the limitations associated with each data collection methodology.

5.2.1 LTE Network Architecture

Internet access in an LTE network is available through base stations (known as eNodeBs) operated by the network provider. The user equipment (UE), such as smartphones, tablets, or LTE modems, connects to the eNodeB over the radio link.

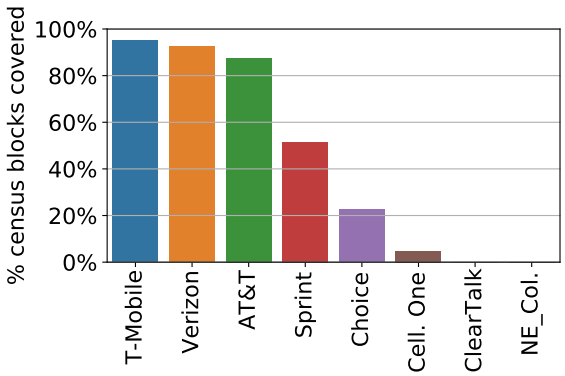


Figure 5.1: LTE operators by census block coverage based on FCC data.

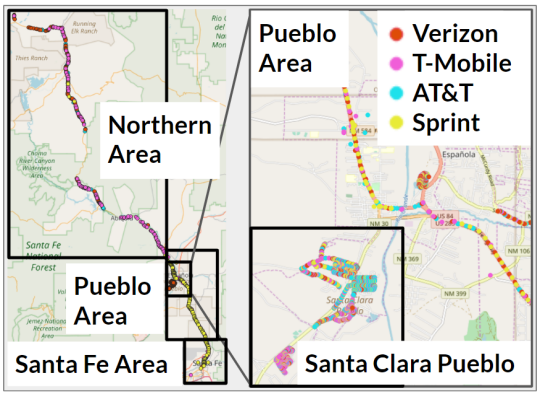


Figure 5.2: Map of author wardriving areas in New Mexico.

The eNodeB is connected to a centralized cellular core called the Evolved Packet Core (EPC). This connection is typically through a wired link forming a middle-mile connection. The EPC consists of several network elements, including a Packet Data Network Gateway (PGW), the connecting node between an end-user device and the public Internet. Thus, LTE broadband access depends on multiple factors, including radio coverage, middle-mile capacity, and interconnection links with other networks (e.g., transit providers, content providers) on the public Internet. However, this article focuses on understanding the last-mile LTE connectivity characterized by the radio coverage of the eNodeB.

An eNodeB controls a single cell site and consists of several radio transceivers or cells mounted on a raised structure such as a mast or a tower. The radio cells use directional antennas, where each antenna provides coverage in a smaller geographical area using one frequency band. The radio cells can be identified through a globally unique number called cell identifier (or cell ID), which is also visible to an end-user device in the cell range. The cell ID enables the aggregation of connectivity and signal strength information from multiple UEs connected to the same cell, which

can then be used to estimate the geolocation of a cell along with its coverage (see Section 5.2.3).

5.2.2 FCC Dataset

The FCC LTE broadband dataset consists of coverage maps in shapefile format that depict geospatial LTE network deployment for each cellular operator in the US. The FCC compiles this dataset semi-annually from operators through Form 477. Every operator that owns cellular network facilities must participate in this data collection. The operators submit shapefiles containing detailed network information in the form of geo-polygons, the frequency band used in the polygon, and the minimum advertised upload and download speeds. The methodology used for obtaining these polygons is proprietary to each operator. Ultimately, the FCC publishes only a coverage map representing coverage as a binary indicator: cellular service is either available or not in any location.

We use the binary coverage shapefiles, available on the FCC website from June 2019[4]. Figure 5.1 shows the eight LTE network operators in the state of New Mexico (NM) and the percentage of total census blocks in NM covered by each operator. Note that we use one of the FCC methodologies to report mobile broadband access, wherein a census block is considered covered if the centroid of the census block is covered [130]. In this chapter, we limit our analysis to the top four cellular operators due to their significantly greater prevalence in NM; these operators are also the leading four cellular operators in the United States more broadly.

Limitations: These coverage maps are generated using predictive models that are

[4]At the time of this analysis, data from December 2019 was also available on the FCC website. However, we use data from June 2019 as the other two datasets in our analysis are collected around this period.

proprietary to the operator [9] and not generally reproducible. Furthermore, the publicly available dataset consists of binary coverage, and lacks any performance-related data.[5]

5.2.3 Skyhook Dataset

Skyhook is a location service provider that uses a variety of positioning tools, including a database of cell locations, to offer precise geolocation to subscribed applications [34]. Through apps that subscribe to Skyhook's location services, user devices report network information gathered into anonymous logs and used to improve the localization service further. Through a data access agreement, we can view the *cell location database* consisting of a list of unique cell IDs along with the cell technology (e.g., 3G vs. LTE), estimated location, and the estimated coverage. The database was initially constructed through extensive wardriving but is now managed and updated using measurements gathered by devices using the Skyhook API for localization. The device measurements with the same cell ID are combined to estimate the cell location and coverage in the following manner:

Cell location estimation: A grid-based methodology similar to that proposed by Nurmi et al. [132] is used to estimate the cell tower location. Specifically, Skyhook divides the geographic area into 7 m squares and groups measurements in the same square to obtain a central measure of the square's signal strength. This is done to reduce the bias due to large numbers of measurements from the same area (e.g., a popular gathering place). A weighted signal strength average is then used to estimate the cell location.

Estimation of cell coverage radius: Skyhook also estimates the cell's coverage radius

[5]The FCC has only recently (beginning December 2019) started giving speed data along with coverage information.

using a proprietary method based on the path-loss gradient [133]. The path-loss gradient approximates how the wireless signal attenuates as a function of the distance from the transmitter (radio cell in this case). The value of the path-loss gradient depends on several factors such as environment (foliage, buildings), geographic topography, and cell signal frequency. Skyhook estimates the path-loss gradient using field observations of cell signal strength readings and their distributed geographic locations. Ideally, the signal attenuation varies based on the direction and the distance from the cell. However, to reduce the complexity of coverage estimation, Skyhook's cell coverage estimation heuristic calculates only one path-loss gradient for a single cell. The path-loss gradient is then used in parameterized equations to estimate the cell coverage radius. The parameters in these equations have been determined with careful research and testing over more than ten years.

The cell location database is regularly updated by recalculating cell location and coverage radius using the new device measurements collected since the last update. For our analysis, we use the cell location database last updated on June 10, 2019.
Limitations: Since database entries are crowdsourced when the device passes within range of a cell, this dataset is more comprehensive in population centers and highways where people often occupy. If there are too few measurements overall, or if measurements are primarily sourced from the same grid section, then the cell location estimate can be inaccurate.

5.2.4 OpenCellID Dataset

Unwired Lab's OpenCellID[6] project provides a publicly available dataset of cell IDs along with their estimated location. The dataset is derived from crowdsourced

[6]OpenCellID Project is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

UE signal strength measurements similar to Skyhook. However, in this case, the UE measurements come from users voluntarily installing the OpenCellID application on their smartphone [129] and manually choosing what data to upload. We differentiate this voluntary crowdsourcing method of data collection from Skyhook's incidental crowdsourcing method, where users of the Skyhook API contribute to the data by default.

Our analysis shows OpenCellID's collection methodology leads to a sparser dataset (see Section 5.3.1).

Another difference between OpenCellID and Skyhook is the methodology used to estimate the cell location. OpenCellID simply assigns the average location of all signal strength measurements corresponding to a single cell location as the estimated cell location. This makes it much more susceptible to error if the measurements are not uniformly distributed. Also, unlike Skyhook, OpenCellID does not estimate the coverage radius of the cells.

Cell information is updated whenever users pass by and gather new information. To increase data accuracy in our OpenCellID analysis, we select only cells that have been updated since January 1, 2018. A single sample represents a device encountering a cell for a single continuous interaction. Even in large counties, almost 40% of all cells have fewer than ten samples. The median number of samples for all large, small, and micro counties considered together are 9, 6, and 4, but standard deviations are 78.3, 94.1, and 90.2. This significant standard deviation suggests extreme outliers in OpenCellID's sampling methodology.

5.2.5 Targeted Measurement Campaign

To complement these datasets, we performed a targeted measurement campaign collecting coverage information through 120 miles of Rio Arriba County in New Mexico for five days beginning May 28, 2019. Figure 5.2 shows the locations of ground measurements and the four descriptive area labels we use for this analysis. The North area measurements were taken on highways passing primarily through the national forest. The Pueblo area measurements were taken from highways within tribal jurisdiction boundaries. In Santa Clara Pueblo, tribal leadership permitted us to collect additional measurements in residential zones. Finally, the Santa Fe area consists of highway measurements between the pueblos and downtown Santa Fe. While limited in scale, these active measurements provide a meaningful comparison point for coverage and user experience. As described in Section 5.1, we selected these areas of New Mexico for their mix of tribal and non-tribal demographics; tribal lands tend to have the highest coverage over-statements and the most limited cellular availability within the United States [6].

Our measurements consist of *service state* and signal strength readings recorded on four Motorola G7 Power (XT1955-5) phones running Android Pie (9.0.0). *Service State* is a discrete variable indicating whether the phone is connected to a cell. Measurements were collected using the *Network Monitor* application [79]. An external GlobalSat BU-353-S4 GPS connected to an Ubuntu Lenovo ThinkPad laptop gathered geolocation tags that were matched to network measurements by timestamp. Each phone was outfitted with a SIM card from one of the four top cellular operators in the area: Verizon, T-Mobile, AT&T, and Sprint. The phones recorded service state and signal strength every 10 seconds while driving at highway speeds (between 40 and 65 miles per hour) in most places and less than 10 miles per hour in residential

County classification	Region	County Name	Population density (per sq. mile)	Skyhook CIDs (#)	Skyhook % Overlap	OpenCellID CIDs (#)	OpenCellID % Overlap	Common CIDs
Large Metro	Western	Los Angeles, CA	2,490.3	133,484	28%	39,875	92%	36,816
	Central	Denver, CO	4,683.0	11,061	24%	3,136	86%	2,689
	Eastern	Fulton, GA	1,994.0	27,809	22%	7,225	86%	6,194
Small Metro	Western	Imperial, CA	43.5	1,818	17%	336	93%	311
	Central	Doña Ana, NM	57.1	1,870	32%	663	89%	592
	Eastern	Bibb, GA	613.0	1,953	21%	464	89%	413
Micropolitan	Western	Tehama, CA	21.7	733	17%	158	80%	126
	Central	Rio Arriba, NM	6.7	333	8%	30	87%	26
	Eastern	Pierce, GA	61.3	164	9%	21	67%	14

Table 5.2: Characteristics and cell ID (CID) counts in selected counties.

areas (Santa Clara Pueblo).

Limitations: Our wardriving campaign was intensive in terms of human effort, economic cost, and time, making it difficult to scale. The dataset does not capture any temporal variations in coverage as the measurements were collected over a short time. It is possible that driving speed or device configuration affects the measurements, e.g., indicating no coverage when a stationary measurement might have detected coverage [134]. We have no evidence that this occurred, but it might warrant some additional investigation.

5.3 Analysis

In this section, we first evaluate Skyhook as a representative crowdsourced dataset by comparing it with a popular *voluntary* crowdsourced data from OpenCellID [129]. This is followed by comparing coverage across the FCC, Skyhook, and our wardriving measurement data. Our comparison is guided by the following questions: (i) what is the degree of coverage agreement across the datasets, ii) where and how do their coverage estimations differ?

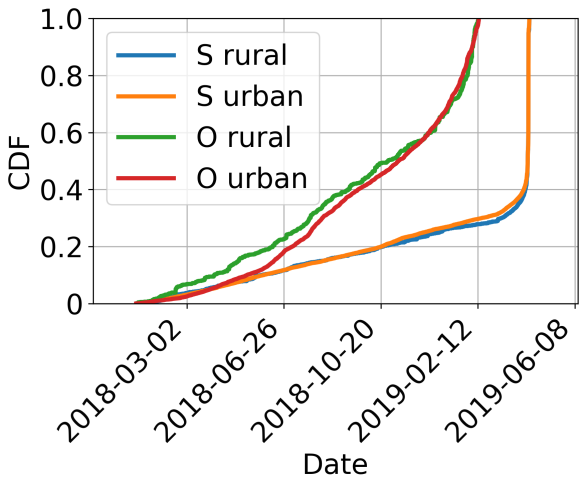


Figure 5.3: CDF of cell updates in Skyhook (S) and OpenCellID (O).

5.3.1 Comparison of Crowdsourced Datasets

We compare the Skyhook dataset with a publicly available crowdsourced dataset – OpenCellID. Unwired Lab's OpenCellID[7] project provides a publicly available dataset of cell IDs and their estimated location. The dataset is derived from crowdsourced UE signal strength measurements similar to Skyhook. However, in this case, the UE measurements come from users voluntarily installing the OpenCellID application on their smartphone [129] and manually choosing what data to upload. We differentiate this *voluntary* crowdsourcing method of data collection from Skyhook's *incidental* crowdsourcing method, where users of the Skyhook API contribute to the data by default. We specifically compare the number of unique LTE cells and the recentness of the measurements in both datasets. We consider each of these factors to contribute to the overall density of the dataset.

methodology: While our coverage comparison will be focused on New Mexico, we analyze our selected crowdsourced data more broadly by considering these datasets

[7]OpenCellID Project is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

within a set of counties of differing population densities across the United States. The counties are selected from three areas of the United States: Western (California), Central (New Mexico and Colorado), and Eastern (Georgia). Within each region, we consider three different kinds of counties as defined by the National Center for Health Statistics' 2013 Urban-Rural Classification Guide [135]. These are: (i) *large metropolitan* (large), which contain a population of at least one million and a principle city; (ii) *small metropolitan* (small), which contain a population of less than 250,000; and (iii) *micropolitan* (micro), which must have at least one urban cluster of at least 10,000, but a total population of less than 50,000. This enables us to study population density and geographic region differences for the crowdsourced datasets. We select three counties of each population category, for a total of nine counties, to compare these two datasets. We describe these counties in Table 5.2. For each county, we show the 2018 population density estimated from the US Census Bureau's 2010 census records [136]. We first count the unique cell IDs in both datasets for each county, as shown in Table 5.2. The "% Overlap" column in Table 5.2 shows the percentage of each dataset's cell IDs that also appear in the other dataset, and the "Common CIDs" column shows the exact number of common cell IDs.

Results: Overall, Skyhook reports a greater number of cells (2.8x - 11.1x) for all counties. The difference is particularly pronounced in micro counties. This suggests that relying on volunteers to download an application and offer network measurements may not be the most accurate method for assessing LTE coverage in rural areas. Furthermore, Skyhook includes a majority of the cells that appear in OpenCellID.

We consider how recently each cell ID record was updated with a new measurement. Figure 5.3 shows the CDF of the latest measurement date for cells in both the datasets, where cells are split into those located in urban and rural census blocks. Almost 60% of the cells in Skyhook were last updated in June 2019, but the most re-

Census block type	Total census blocks	Verizon		T-Mobile		AT&T		Sprint	
		FCC	Skyhook	FCC	Skyhook	FCC	Skyhook	FCC	Skyhook
Non-Tribal Rural	93,680	89%	77%	94%	86%	85%	79%	39%	49%
Non-Tribal Urban	41,872	100%	100%	100%	100%	99%	99%	96%	99%
Tribal Rural	30,588	93%	80%	92%	63%	78%	73%	27%	41%
Tribal Urban	2,469	100%	99%	95%	94%	93%	94%	75%	88%
All	168,609	93%	84%	95%	85%	88%	83%	52%	61%

Table 5.3: Percentage of total census blocks covered according to FCC and Skyhook.

Block type	Total blocks	Verizon	T-Mobile	AT&T	Sprint
Non-Tribal Rural	93,680	14,013	9,025	8,705	1,355
Non-Tribal Urban	41,872	0	0	213	25
Tribal Rural	30,588	5,109	9,150	3,004	230
Tribal Urban	2,469	4	14	4	0

Table 5.4: Number of census blocks where there is coverage according to FCC but no coverage according to Skyhook.

cent update in OpenCellID was in February 2019. Furthermore, cells in rural census blocks were updated less recently than urban census blocks in OpenCellID, while the difference is negligible in the Skyhook dataset. This suggests that the Skyhook dataset is updated more regularly than OpenCellID, thus making it more likely to represent any changes in the network infrastructure.

5.3.2 Comparison of Coverage

Coverage comparison between the FCC and Skyhook

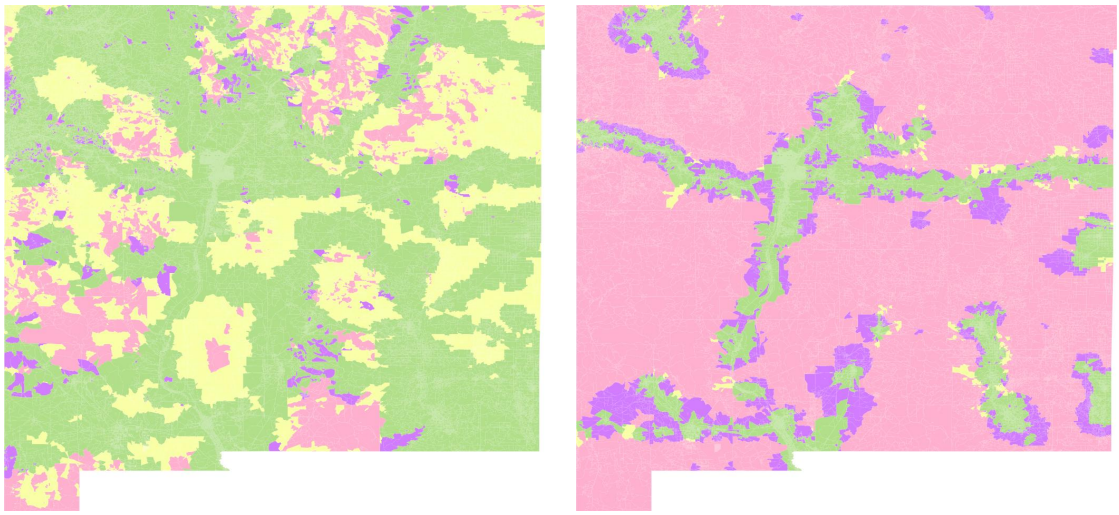
We first compare a coverage shapefile generated from Skyhook cell locations and estimated coverage ranges with the FCC map for each operator.

Methodology: We consider coverage at the census block level for this comparison. In addition to reporting coverage shapefiles, the FCC reports coverage at a census block level and considers a census block as covered if the centroid of the census block falls within a covered region [130]. Using Skyhook's estimated coverage, we generate a similar census block level coverage map per operator. To do so, we first obtain the

coverage shapefile for each operator using a cell's estimated location and coverage radius. Then we use the FCC centroid methodology to generate the Skyhook LTE coverage map at the census block level. We use the Python GeoPandas 0.8.2 library for the associated spatial operations [137]. We group census blocks into four categories: Non-Tribal Urban, Non-Tribal Rural, Tribal Urban, and Tribal Rural. This is done to explore whether the degree of agreement of the two datasets varies across these dimensions. We use the US Census Bureau's classification of urban and rural blocks and its boundary definitions of tribal jurisdiction for this categorization [138]. In this analysis, we consider census blocks as tribal if they overlap with tribal boundaries. We varied the tribal labeling schemes, such as classifying a census block tribal if the centroid of the block is within a tribal boundary. However, the results remain qualitatively similar and do not impact the findings.

Results: Table 5.3 shows the percentage of total census blocks covered by each cellular operator, according to the FCC and Skyhook data, broken down by census block type. Among the four operators, T-Mobile covers the greatest number of census blocks based on both FCC and Skyhook data, while Sprint covers the fewest. All four cellular operators have relatively higher coverage for tribal and non-tribal urban census blocks. However, all operators except Verizon offer their lowest coverage in tribal, rural areas. For some operators, the differences between non-tribal rural and tribal rural are as great as 23% (based on Skyhook data) and 11% (based on FCC data).

The extent of LTE coverage differs between the two datasets. For three out of four providers, Skyhook shows lower coverage than the FCC, particularly in the rural census blocks. For instance, the FCC T-Mobile data shows coverage in 92% of tribal and rural blocks, whereas Skyhook shows coverage in only 63% of such blocks. On the other hand, Skyhook shows more census blocks covered than the FCC for Sprint.



a. Verizon b. Sprint

Figure 5.4: Comparison of LTE coverage maps of New Mexico. Yellow blocks are covered in the FCC map but not in Skyhook; purple blocks are covered in the Skyhook map but not the FCC. Green blocks are covered in both, and pink blocks are covered in neither.

The higher coverage in the case of Sprint could have been due to multiple reasons, including (i) there are differences in the propagation models used by Skyhook and Sprint to estimate coverage, with the former's models being more generous than the latter's, and (ii) the Skyhook data is collected across time, and Sprint may have discontinued or temporarily disabled some of the cells, which is challenging to detect from the crowdsourced data.

Figure 5.4 visually compares the LTE coverage maps from the FCC and the Skyhook datasets for Verizon and Sprint. We more deeply examine the discrepancy mapped in yellow in Figure 5.4(a). Table 5.4 shows the number of census blocks with coverage according to the FCC but none according to Skyhook for each operator. Coverage claims in both tribal and non-tribal rural census blocks disagree the most. The number of such blocks are particularly high for Verizon (19,126 overall) and T-Mobile (18,189 overall). There are two possible reasons for this disagreement:

Block type	Verizon	T-Mobile	AT&T	Sprint
Non-Tribal Rural	528 (1%)	2,575 (3%)	5,342 (6%)	19 (<1%)
Non-Tribal Urban	0 (0%)	0 (0%)	213 (1%)	0 (0%)
Tribal Rural	2,655 (9%)	2,565 (8%)	2,166 (7%)	0 (0%)
Tribal Urban	0 (0%)	0 (0%)	4 (<1%)	0 (0%)

Table 5.5: Number of census blocks with LTE coverage according to the FCC, but only 3G coverage according to Skyhook. The numbers in parenthesis report the same data as a percentage of total census blocks of the corresponding type.

network operators lack adequate infrastructure in rural areas but tend to overestimate coverage while reporting it to FCC, or Skyhook is missing data points from rural census blocks where fewer people carry UEs/ The latter case will lead to either some LTE cells not being detected or an inaccurate characterization of cell coverage due to fewer measurements.

To understand which of these potential reasons for disagreement is more likely, we check whether Skyhook shows 3G coverage for these census blocks (where the FCC reports LTE coverage but Skyhook does not). If Skyhook reports 3G coverage in these blocks, this suggests that users may have contributed to the Skyhook dataset in these census blocks. Therefore, LTE coverage would have been detected if it existed. Note that a more accurate way would have been to directly consider the location of end-user measurements connected using 3G technology and analyze whether they fall within LTE coverage areas in the FCC data. However, we did not have access to these end-user measurements due to Skyhook's privacy policy. Instead, we consider the 3G coverage maps as a reasonable approximation for our analysis and generate a 3G coverage map at the census block level for these areas, in the same manner described previously. The number of census blocks that show only 3G coverage according to Skyhook is presented in Table 5.5. We observe a significant number of census blocks where Skyhook detects 3G coverage, indicating that the FCC LTE coverage claims may be overstated in these areas. The number of such blocks is

greater for tribal, rural areas (up to 9%), thus indicating a higher mismatch of the two datasets in tribal, rural areas.

Active measurements compared to FCC and Skyhook coverage

In this section, we compare our active measurements with the coverage maps from the FCC and Skyhook described in Section 5.3.2. We focus now on the geographic region around Santa Clara Pueblo, which lies north of Santa Fe (see Figure 5.2), a region with a mix of urban, rural, and tribal population blocks.

Methodology: We use the *Service State* readings collected in our measurements for this analysis (see Section 5.2.5). We also collected information about the connected cell's technology (e.g., LTE) and the geolocation of the measurements. This information is used to infer whether LTE coverage exists at a location. We consider LTE to be available if the *Service State* shows IN_SERVICE to indicate an active connection and if the associated cell is an LTE cell. We term this the *active* LTE coverage. We then compare the FCC and Skyhook coverage with the active LTE coverage to see whether the datasets agree. Note that we use the coverage shapefiles for Skyhook and the FCC in this comparison instead of the census block centroid approach in Section 5.3.2. This allows us to compare coverage more precisely for a location, especially if a census block is only partially covered.

Results: Table 5.10 shows the confusion matrices that compare active LTE coverage with reported coverage from the FCC and Skyhook maps. Both maps show coverage at locations where our measurements did not. In the case of Verizon, 81% of the measurements with no coverage are from locations reported as covered by the FCC. This over-reporting is lowest for Sprint and highest for T-Mobile.

We also observe significant disagreement (up to 79%) between Skyhook coverage and our measurements. Two possibilities may cause this: i) lack in Skyhook UE

Active	Total	FCC		Skyhook	
		NC	C	NC	C
No Coverage (NC)	266	19%	81%	32%	68%
Coverage (C)	1,440	0%	100%	5%	95%

Table 5.6: Verizon

Active	Total	FCC		Skyhook	
		NC	C	NC	C
No Coverage (NC)	324	6%	94%	21%	79%
Coverage (C)	1,361	0%	100%	5%	95%

Table 5.7: T-Mobile

Active	Total	FCC		Skyhook	
		NC	C	NC	C
No Coverage (NC)	568	25%	75%	53%	48%
Coverage (C)	1,095	2%	98%	7%	93%

Table 5.8: AT&T

Active	Total	FCC		Skyhook	
		NC	C	NC	C
No Coverage (NC)	231	96%	4%	99%	2%
Coverage (C)	1,122	21%	79%	20%	80%

Table 5.9: Sprint

Table 5.10: Confusion matrices comparing active measurement coverage with FCC and Skyhook. *Total* denotes the number of active measurements in each category.

signal strength readings available for cell location and coverage radius estimation, or ii) error in the cell propagation model itself, possibly due to variations in the environmental conditions such as the terrain. In either case, Skyhook agrees better with our measurements than the FCC in reporting areas with no LTE coverage. E.g., in the case of AT&T, 75% of our measurements with no LTE coverage belong to areas reported as covered by the FCC as compared to 48% by Skyhook.

5.4 Recommendations

This section discusses some of the implications of our experience collecting and analyzing coverage data, recommendations based on our findings, and directions for future work.

Recommendations for the FCC: Our findings make a case for including mechanisms that validate ISP-reported coverage data, especially in rural and tribal regions. Given the scale of cellular networks, crowdsourcing coverage measurements is a viable approach to validate access as opposed to controlled measurements. Within crowdsourcing, we suggest leveraging *incidental* rather than *voluntary* approaches, possibly working with third-party services that collect network measurements as part of their service process (as in the case of Skyhook).

In addition, crowdsourcing alone may not be sufficient for determining coverage in some cases. Even with the more complete datasets provided through incidental crowdsourcing, rural areas tended to receive significantly fewer measurements per tower. In such cases, mechanisms need to be developed to determine areas of greatest disagreement using sparse crowdsourced datasets. Resources can then target data collection in these areas instead of a blanket approach measuring coverage everywhere.

Recommendations for crowdsourced data collection: We find some shortcomings in the existing crowdsourced datasets. First, existing datasets only report areas with favorable coverage, i.e., areas where coverage is observed. This makes it challenging to distinguish areas lacking coverage from areas where no measurements were gathered. Recording areas that lack a usable signal can enable more decisive conclusions from crowdsourced data.

Second, we note that even crowdsourced datasets are prone to overestimating

coverage potentially due to errors in cell location and coverage estimation. Research efforts that effectively utilize the knowledge of cellular network design are needed for the accurate characterization of coverage from crowdsourced measurements. For instance, existing cell location estimation techniques localize cells independently (see Section 5.2.3) and are prone to errors when there are few end-user measurements [139]. Instead, one can utilize the fact that a single physical tower in an LTE network hosts multiple cells. Thus, algorithms that jointly localize cells for whom the end-user measurements are in physical proximity may provide higher accuracy even with fewer end-user measurements. Similarly, alternate data sources can also be considered for localizing cell infrastructure, such as using geo-imagery data to identify physical towers or directly obtaining infrastructure data from entities that build and manage physical cell towers (usually different from cellular ISPs).

Measuring access beyond binary coverage: While this work focuses on understanding coverage, we recognize that a binary notion of coverage alone does not necessarily indicate the existence of functional LTE connectivity. Various other factors can impact end-user experience in “covered" areas, such as low signal strength or *poor* middle-mile connectivity. Thus, future coverage measurement efforts need to augment coverage reports with performance measurements to provide models more aligned with user experiences. Measuring such performance metrics poses a greater challenge because end-user experience depends on many factors beyond just last-mile link quality. We believe that efforts that lead to increased community awareness (e.g., workshops in public libraries and community meetings) on the importance of measuring mobile coverage are the way to tackle this problem.

We note that access and adoption are different, and issues beyond access might also warrant measurement and consideration as accountability measures for operators. Our collection of ground truth data sets involved five days of driving through

Rio Arriba County in northern New Mexico. In preparation for the trip, we worked to obtain SIM cards that would enable us to access the networks of the four major U.S. LTE operators. This was surprisingly difficult; for a month leading up to the measurement campaign, we spent 24 hours in various operator kiosks and stores in three states to obtain eight SIM cards (one for each major operator). At one of the stores in Santa Fe, we encountered a woman who had to drive an hour from Las Vegas, NM, to address some of the issues she was having with her mobile service operator that were preventing her from using her data plan. While these anecdotal experiences mirror the qualitative claims of coverage overestimation, they introduce a new set of issues that need to be taken into account to reduce the barriers to Internet access for rural communities effectively.

Finally, we observe that there are issues beyond availability that influence access that might also warrant measurement and consideration as accountability measures for operators. Our collection of ground truth data sets involved five days of driving through Rio Arriba County in northern New Mexico. In preparation for the trip, we worked to obtain SIM cards that would enable us to access the networks of the four major U.S. LTE operators. This was surprisingly difficult; for a month leading up to the measurement campaign, we spent 24 hours in various operator kiosks and stores in three states to obtain eight SIM cards (one for each major operator). At one of the stores in Santa Fe, we encountered a woman who had to drive an hour from Las Vegas, NM, to address some of the issues she was having with her mobile service operator that were preventing her from using her data plan. While these anecdotal experiences mirror the qualitative claims of coverage overestimation, they do introduce a new set of issues that need to be considered for

5.5 Conclusion

In this chapter, we quantitatively examine the LTE coverage disagreement among existing datasets collected using different methodologies. We find that existing datasets display the most divergence when compared with each other in rural and tribal areas. We discuss our findings for their implications for telecommunications policy. We also identify several future research directions for the computing community, including mechanisms to augment existing datasets to precisely determine areas where more concerted measurement efforts are needed, improved coverage estimation models, especially for areas with a lower density of crowdsourced measurements, and accurate and scalable measurement of access beyond a binary notion of coverage. For the first three chapters of this thesis, the evaluations were strictly based on mobile broadband. In the next chapter, we study Internet accessibility on fixed broadband through the lens of the transport layer protocols.

Chapter 6

MPTCP Performance over Heterogeneous Subpaths

6.1 Introduction

The challenge to meet 5G throughput goals has motivated today's mobile devices to use Wi-Fi and cellular networks simultaneously. To take advantage of multiple available networks, researchers have created Multipath TCP (MPTCP), a transport protocol designed to achieve better throughput and resource utilization by enabling the simultaneous use of several IP addresses/interfaces [140]. A unique challenge facing widespread MPTCP usage is the considerable performance differences commonly observed between cellular and Wi-Fi networks [141].

Prior work has explored the performance of MPTCP in a variety of contexts. While much of that work has focused on implementing and measuring MPTCP performance, few studies have focused on the default MPTCP scheduler, especially from the perspective of heterogeneous subpaths. Today, path heterogeneity plagues the performance of various networks, such as Wi-Fi and cellular networks. In addition,

cellular networks traditionally have significantly higher latency and loss compared to Wi-Fi [141]. As studied by [142], single-path TCP performs better than MPTCP over HTTP/2, especially when page size is small or network transfer is not the bottleneck. Furthermore, mobile users should not enable Multipath TCP without thoughtful consideration because it may lead to higher energy and cellular data consumption without providing significant quality of experience (QoE) improvements [142]. This study aims to understand MPTCP performance on paths with differing characteristics. In particular, we focus on paths of varying latency and loss rates and assess MPTCP performance in these contexts. We seek to answer the following questions:

* *What is the impact of choosing a particular interface over another to act as the primary, particularly when DNS resolution is a concern?*
* *How does interface selection impact MPTCP availability and server reachability?*
* *How does MPTCP perform when paths have differing performance characteristics?*
* *When is single path TCP a better choice than MPTCP?*

We begin our study by measuring packet round-trip times (RTTs) for the Tranco top 10K websites [36] over different interfaces: Wi-Fi and LTE. RTT is a crucial parameter in network performance; the default MPTCP scheduler makes path selection decisions that rely heavily on the RTT of each path. Different interfaces can produce vastly different RTTs for the same web servers, partly due to server replication within CDNs. Next, server reachability is critical in assessing whether MPTCP can be implemented in a broad range of scenarios. Additionally, it helps us understand the importance of selecting the primary interface. Informed by our findings, we perform controlled in-lab experiments that vary network latency and loss conditions to examine the effect on metrics such as download times and page load time. Lastly,

we replicate the in-lab test conditions to investigate MPTCP performance in the real world.

Our experiments indicate that selecting cellular as the primary interface can be detrimental because of two reasons: a) RTT observed on the cellular path can be considerably higher than on Wi-Fi, and b) IP addresses resolved on the cellular interface have a lower chance of reachability than those resolved on Wi-Fi. We show empirical data that the default MPTCP scheduler can underperform when paths are disproportionately lossy. We compare MPTCP with single-path TCP to study how small flows suffer more in comparison.

The rest of the chapter is organized as follows. We provide a brief introduction to MPTCP and discuss related work in Section 6.2. In Section 6.3, we present our methodology and results from the real-world survey. Section 6.4 describes our controlled experiment testbed, methodology used, and results. Real-world test results are then presented in Section 6.5. Finally, we conclude our analysis in Section 6.6.

6.2 Background

This section presents a brief overview of MPTCP and the default scheduler. We then discuss related work to understand how our study explores a less-analyzed aspect of MPTCP performance.

6.2.1 Multipath TCP Overview

MPTCP is a TCP extension that enables concurrent data transmission from one end-to-end connection over multiple paths. For instance, on a smartphone, MPTCP allows applications to simultaneously send and receive data over multiple interfaces,

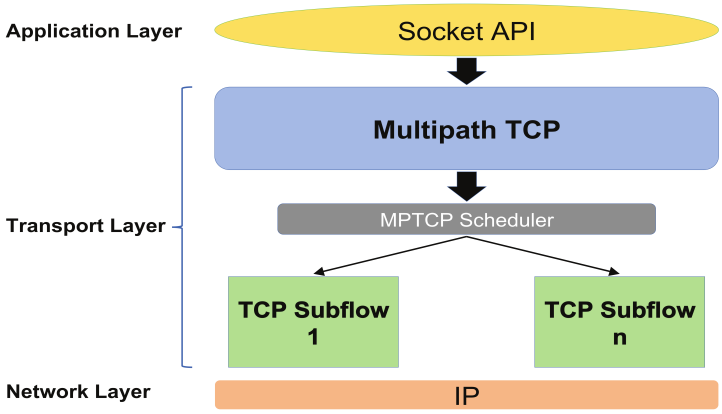


Figure 6.1: Multipath TCP architecture.

such as Wi-Fi and cellular, by establishing one TCP subflow over each interface [140]. Once a subflow is established, it can be used by the MPTCP scheduler to transmit data. MPTCP provides various benefits, including better resource utilization, higher throughput, and smoother reaction to failures. It is likely to work better with paths with comparable qualities. Figure 6.1 shows the architecture of Multipath TCP.

When multiple subflows are available to send data, the default scheduler [143] will transmit the data on the subflow with the shortest round trip time. When a segment is ready to be transmitted, the scheduler chooses the path with the minimum RTT out of all subflows whose congestion window is not yet full. Suppose there is more than one such path. In that case, the scheduler develops a systematic inclination towards one of the interfaces and continues to transmit data on that particular subflow until the subflow's congestion window becomes full. The interested reader can learn more about MPTCP in RFC 6824 [140].

6.2.2 Related Work

Previous work has examined MPTCP in a mobile context. For instance, [144] studied the impact of mobility on MPTCP, while [145] proposed different MPTCP

modes to be used by mobile devices for cellular/Wi-Fi handover. However, neither work explored path heterogeneity with regard to lossy subflows. [146] presented a measurement study that compares single-path TCP to MPTCP. Arzani et al. [147] studied the effect of the scheduler design on performance by using different scheduler algorithms. In contrast, others have compared various congestion control algorithms [148][149][150][151]. Little prior work studying the effect of selecting the primary resolution interface exists. Yang et al. [152] proposed an alternative scheduler that chooses subflows based on estimating how much more traffic they can handle before becoming congested. Their approach considers scenarios with very large transfers in a network with a small amount of buffering. Another scheduling algorithm was proposed in [153] to avoid out-of-order segments. However, the authors do not explain how to remove a segment from a TCP buffer once it is retransmitted from another subflow.

Kuhn et al. [154] proposed a delay-aware packet scheduler, which is evaluated only through ns-2 simulations. Their method examines path heterogeneity in stable CWND and delay conditions only. Closest to our work is [155], which measured MPTCP performance over cellular networks and Wi-Fi. This study focuses on varying numbers of subflows and detailed statistics, such as out-of-order delivery and round trip times, but does not consider lossy subflows. Lim et al. proposed a scheduler that monitors the available bandwidth on each subflow and send buffers; however, it does not take advantage of the loss rate information on each subflow.

6.3 Latency and Reachability Survey

The MPTCP scheduler relies on round rip latencies for path selection. Hence, it is imperative to understand the typical RTT differences between interfaces in a

mobile context. Furthermore, MPTCP can be impossible to implement when the server IP address is unreachable over one of the subpaths. We study these parameters in our real-world survey to understand the obstacles MPTCP faces for widespread deployment.

6.3.1 Methodology

Since the default MPTCP scheduler relies heavily on round-trip times for path selection, our first goal is to study the RTT difference of each potential path to the path's corresponding web server. To do so, we survey to explore latency and reachability for each of these interfaces to the Tranco top 10K websites [36]. Our testbed consists of a Samsung Galaxy S5 phone tethered to a Lenovo Thinkpad laptop via USB. We access the Wi-Fi network through the tethered phone instead of the built-in interface in the laptop to ensure we account for the performance overhead added through tethering and maintain consistency across all the interfaces. We use the T-Mobile network for LTE services. The tests were performed after midnight to avoid high-usage times, and we ensured the signal strength on the devices was strong. Latencies were collected through Hping3 by averaging the RTT of 10 packets sent to each target IP address through each interface.

6.3.2 Results

As the first step in our study, we perform DNS resolution of the Tranco top 10K websites [36] on each interface (Wi-Fi and LTE) using the tethered phone. Web servers can resolve to different IPs over different interfaces. This is because resolution depends on how the ISP routes the request to return the address of the desired content delivery network (CDN). For instance, cellular operators will likely embed web

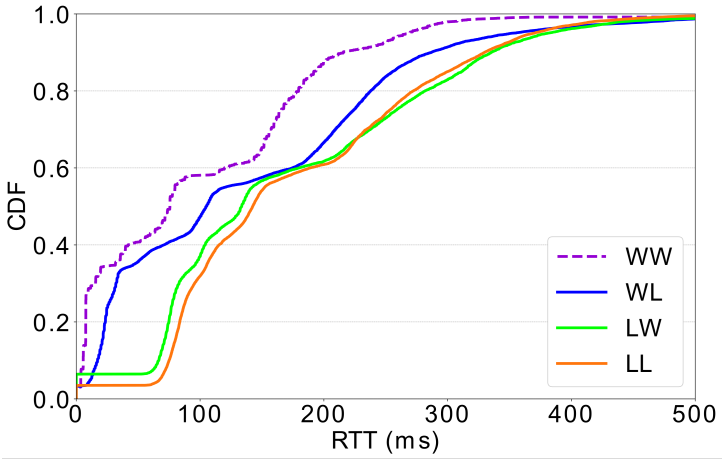


Figure 6.2: CDF of Tranco top 10K web-servers' RTT, resolved using different interfaces.

servers and CDNs within their core network to provide faster response times to user web requests. Omitting websites that do not resolve, the resulting sample size after DNS resolution is 9756 and 9638 for Wi-Fi and LTE, respectively, with an overlap of 58.74% in IP addresses. We then conduct RTT tests on the obtained web servers using Hping3. Hping3 uses TCP packets to ping the servers. There are four possible DNS resolution/latency combinations. They are:

1. ***WW***: Wi-Fi interface using address resolved on Wi-Fi DNS
2. ***WL***: Wi-Fi interface using address resolved on LTE DNS
3. ***LW***: LTE interface using address resolved on Wi-Fi DNS
4. ***LL***: LTE interface using address resolved on LTE DNS

Figure 6.2 shows the cumulative distribution function (CDF) for the average round-trip time for 10 TCP packets sent to each target server collected over each interface. We observe that the RTT deviation between *WW* and *WL* is around 40ms, while the mean RTT deviation is about 75ms between *WW* and websites accessed through LTE

Ping Interface	Resolution Interface	Percentage
Wi-Fi	LTE	**4.57**%
Wi-Fi	Wi-Fi	3.26%
LTE	LTE	3.50%
LTE	Wi-Fi	3.59%

Table 6.1: Percentage of unresponsive servers.

(*LL* and *LW*). It is interesting to note that in about 60% of websites, *LL* under-performs in comparison to *LW*. This result is quite surprising: web servers resolved over LTE, likely CDN servers within the cellular network infrastructure, incur more significant delays. We speculate that this happens because servers become unresponsive soon after DNS resolution. Possible reasons for this behavior can be attributed to load balancing across several IP addresses associated with a web server, and continual Denial of Service attacks [156][157]. In addition, LTE should be able to access the servers resolved on Wi-Fi. It is clear from Figure 6.2 that the use of the web servers resolved over the cellular interface will likely yield larger delays and hence adversely affect user experience with longer RTTs.

Table 6.1 shows the percentage of unresponsive servers during the latency test. We define unresponsive as those servers that never send a response back to our pings. We study the reachability of servers to understand whether MPTCP is possible to use in all cases (assuming web servers were MPTCP enabled). If a server is unreachable over a particular interface, which MPTCP uses as one of its subflows, then MPTCP is unusable. In other words, it is no better than using single-path TCP. From Table 6.1 we note that the Wi-Fi interface produces far fewer unresponsive servers than the LTE interface; more than 4.5% of the web servers resolved on LTE are unresponsive. We posit that this is due to unreachable servers deployed in the cellular core behind NAT and interference by middleboxes. On the other hand, servers resolved using Wi-Fi

are more likely to be reachable via other interfaces. Note that more than 40% of the web servers either resolve differently or do not resolve at all depending on the choice of interface. This is noteworthy because there is a high probability that either of the interfaces (i.e., Wi-Fi or LTE) will be unable to establish an end-to-end path with a web server resolved over the other path. The takeaway from these observations is that the end-to-end path is unavailable in the presence of unreachable servers. In other words, MPTCP will be impossible to use.

6.4 Controlled in-lab experiments

As shown in Section 6.3, a considerable disparity exists between the round-trip times to web servers through cellular and Wi-Fi interfaces. This disparity will likely influence the path selection process and the resulting performance. In this section, we study the effect of varying latency and loss rates on performance metrics to indicate how the MPTCP scheduler will perform across heterogeneous links.

6.4.1 Testbed Setup

Figure 6.3 illustrates the testbed used in our second set of experiments, where we study the performance of MPTCP by manually setting loss and latency parameters in our controlled environment. The testbed consists of a wired server and client housed

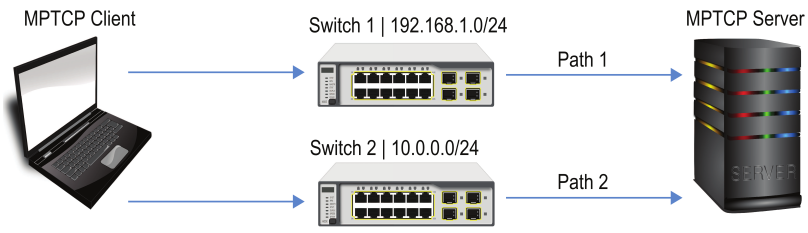


Figure 6.3: Experimental setup.

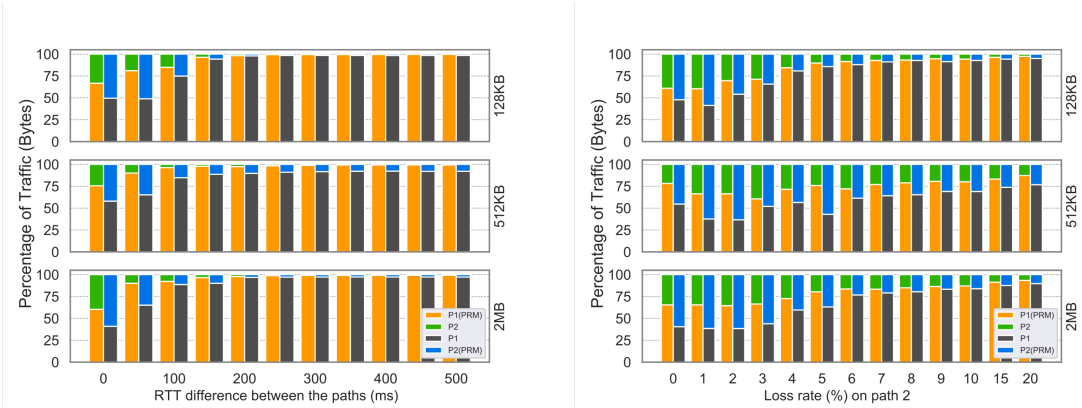
at our research facility. The client is connected to the server through two interfaces via switches, resulting in two paths between the devices. The client and server are both Lenovo ThinkCentre M910T machines (Ubuntu Linux 16.04 with MPTCP Kernel implementation version v0.92) configured with Intel Core i7-7700 processor (3.6GHz) and paired with 64GB of DDR4 RAM. Each machine comes with one integrated Intel Gigabit Ethernet interface. We install an additional TP-Link Gigabit Ethernet card for our experiment on both machines. We disable the wireless interfaces on the client and server. The switches are both Linksys WRT1200AC dual-band routers, running OpenWrt version 15.05.1.

6.4.2 Methodology

The motivation of our controlled experiments is to generate baseline results that can be used as a reference for the real-world experiments in Section 6.5. For in-lab testing, we configure our lab server as an HTTP server, running Apache2 on port 80. We first cache the Tranco top 1000 websites to evaluate page load time on our local lab server. We then establish an MPTCP connection from the client to the server in our testbed, wherein the client fetches each cached website. The experiment is repeated 100 times for each website, and the average page load time is calculated. While this approach does not exactly translate to fetching live webpages, the results serve as a reference for real-world experiments. Hence, our results are an approximation of actual page load times. Note that our testbed does not reflect the true RTT between the client and server. Therefore, the controlled experiments consider a broad range of inter-path RTT differences.

Next, we set up a 2-path MPTCP connection between the client and server in our testbed. The paths are set up on two different subnets. We aim to structure these

paths to emulate two different interfaces on a device, e.g., Wi-Fi and LTE. To study web traffic performance over MPTCP, we choose various file sizes for measurement: 128KB, 256KB, 512KB, 1MB, and 2MB. To study various network conditions, we vary

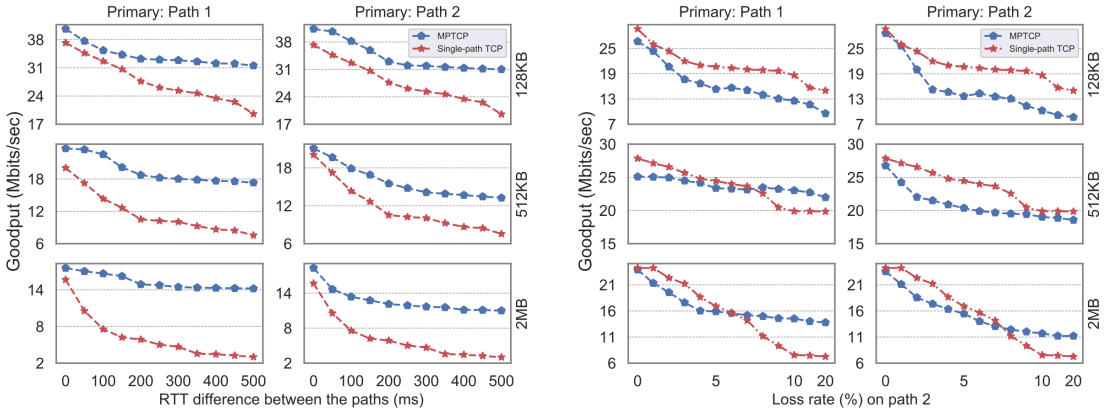


(a) RTT on path 2 increases (b) Loss rate on path 2 increases

Figure 6.4: Fraction of traffic carried on each path for controlled iPerf experiments.

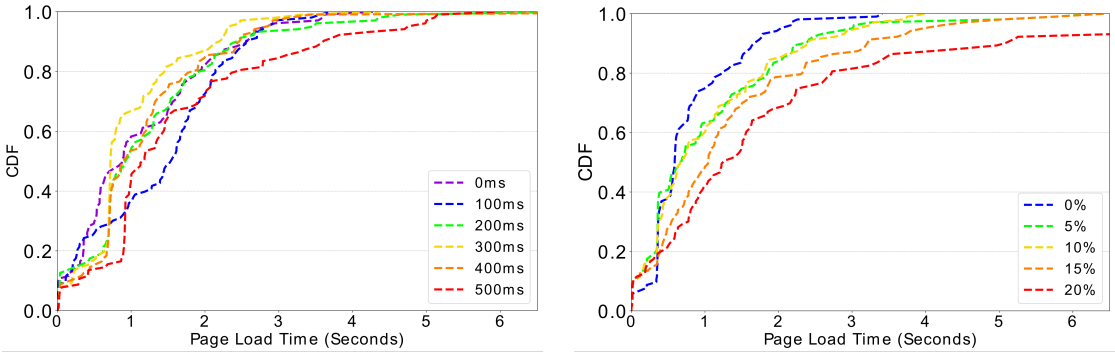
either the latency or the loss rate of path 2, keeping the other parameter constant. We use the Linux Traffic Control `tc` command for this purpose. During our survey in Section 6.3.2, we found the median RTT to be around 20ms and loss rates consistently about 0.1% for Wi-Fi. Therefore, we initiate both the paths with a 20ms round-trip time and 0.1% loss rate. To study the effect of differing path latency, we increase the RTT to 50ms, and in each subsequent experiment, we increase it in increments of 50ms to a maximum of 500ms on path 2. To study the effect of loss rate, we initialize both paths with no loss and 20ms of RTT. After that, we increment the loss by 0.5% on path 2 to a maximum loss rate of 10%. We record measurements for 15% and 20% loss rates as well.

As our goal is to understand the effect on performance when a higher RTT path is chosen instead of the lower RTT path as the primary interface, we initialize these experiments by manually specifying the primary path at the beginning of the experiment. That is, we run two sets of experiments. First, path 1 is set as the primary path,



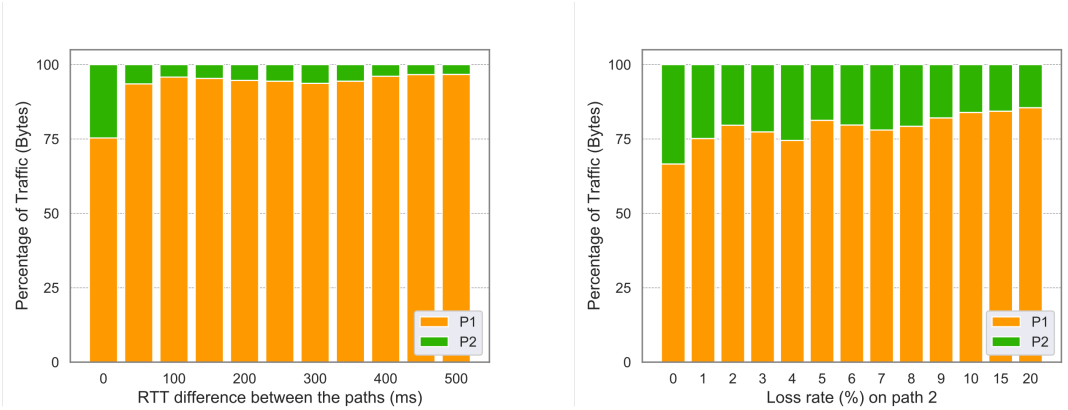
(a) RTT on path 2 increases (b) Loss rate on path 2 increases

Figure 6.5: Achieved goodput for web downloads (higher is better).



(a) RTT on path 2 increases (b) Loss rate on path 2 increases

Figure 6.6: CDF of page load time (lower is better).



(a) RTT on path 2 increases (b) Loss rate on path 2 increases

Figure 6.7: Fraction of web traffic carried on each path for controlled experiments.

and second, path 2 is configured as the primary path. This approach is analogous to MPTCP choosing the primary interface for DNS resolution in a real-world scenario. As we explained in Section 6.3, the primary interface chosen for DNS resolution plays an important role in the reachability of servers. Note that MPTCP can and will, in subsequent stages, choose which path to send traffic through, a decision dictated by the MPTCP scheduler. We measure goodput by running `iperf` [158] in client-server mode. To specify file transfer size, corresponding to our short and long flows, the *-n* flag in `iperf` is assigned accordingly. Each set of experiments is run 100 times for every file size. At the server-side, we capture the traffic using `tcpdump` [159].

6.4.3 Evaluation

In this section, we present the results of our study on increasing RTT and loss rate on the range of traffic sent on each path. Figures 6.4(a) and (b) show the percentage of traffic sent over the primary and secondary paths as latency and loss differ. In Figure 6.4(a), we see that as the RTT increases on path 2, the traffic is directed almost exclusively on the lower RTT path 1. This is because the algorithm with which the

default scheduler initiates path selection gives higher priority to paths with lower RTT [160]. However, in Figure 6.4(b), we see that path 2 still carries traffic, even though the loss rate increases. For short flows like these, the subflows may never exit their slow-start phases, which explains why path 2 is still significantly used.

Figure 6.5(a) shows the achieved goodput for 128KB, 512KB and 2MB file sizes as the RTT on path 2 increases. Due to space restrictions, we omitted flow sizes of 256KB and 1MB from these plots. We see a steady decrease in goodput until an RTT difference between 200ms and 250ms, after which the goodput flattens. A similar pattern is observed in file sizes of 512KB and 2MB. This behavior can be explained by Figure 6.4(a), which indicates that, once the inter-path RTT exceeds 200ms, the bulk of traffic is carried on path 1 due to its lower RTT. It is clear from Figure 6.5(a) that after 200ms, additional increases in RTT on path 2 cease to have any significant effect on the goodput. We observe that MPTCP performs better than single-path TCP on all occasions, regardless of the choice of the primary path. Next, Figure 6.5(b) shows the achieved goodput for increasing loss rate on path 2. When we compare MPTCP with single-path TCP in Figure 6.5(b), we observe that for the short flows of sizes 128KB and 512KB, single-path TCP outperforms MPTCP. For flow sizes this small, subflows can still be in their slow-start phase when the download is complete. Additionally, for the 2MB file size, we observe that TCP performs as well as MPTCP, if not slightly better. This result tells us the importance of the initial path selection process, particularly because a considerable fraction of web traffic is small flows [161].

Page load time directly affects user experience, and is represented in Figure 6.6. Figure 6.6(a) indicates that the page load time does not necessarily increase as the RTT increases on path 2. We observe that as the RTT increases on path 2, the bulk of the traffic is carried on path 1, which is associated with lower latency. However, as we increase the loss rate on path 2 as shown in Figure 6.6(b), page load time gradually

increases. To better understand this result, we study the fraction of traffic carried on each path in Figure 6.7(a). While the vast majority of traffic traverses path 1 as the RTT on path 2 increases, this does not hold true as the loss rate on path 2 increases. In Figure 6.7(b), even though path 2 is lossier, it still carries an appreciable amount of traffic, over 20% in most cases. As a result, page load times slowly degrade as we approach the 20% loss rate on path 2. This is because TCP's estimation of a path's RTT is not affected by the packet loss on that particular path; the MPTCP scheduler ignores RTT for retransmitted and lost packets. Stated otherwise, it does not take into account path loss. Consequently, this adversely impacts path selection and hence performance.

6.5 Real world experiments

We next examine the performance of the default scheduler in a more realistic scenario. This enables us to gauge whether the conditions identified in Section 6.4 occur in practical settings. Wherever possible, we replicate the test cases used in Section 6.4 to establish a fair comparison.

6.5.1 Testbed Setup

For this set of experiments, we deploy two MPTCP-enabled machines on a popular cloud service provider. Our testbed consists of a server located in Virginia and a client that is situated in California. We keep the system configuration similar to our controlled experiments, i.e., Ubuntu Linux 16.04 with MPTCP Kernel implementation version v0.92. The client communicates with the server over the Internet through two different wired interfaces, each connected to a different subnet to maintain isolation

of routes. We observe an average round-trip time of 61ms with a standard deviation of $\pm$3ms between the client and server on both paths.

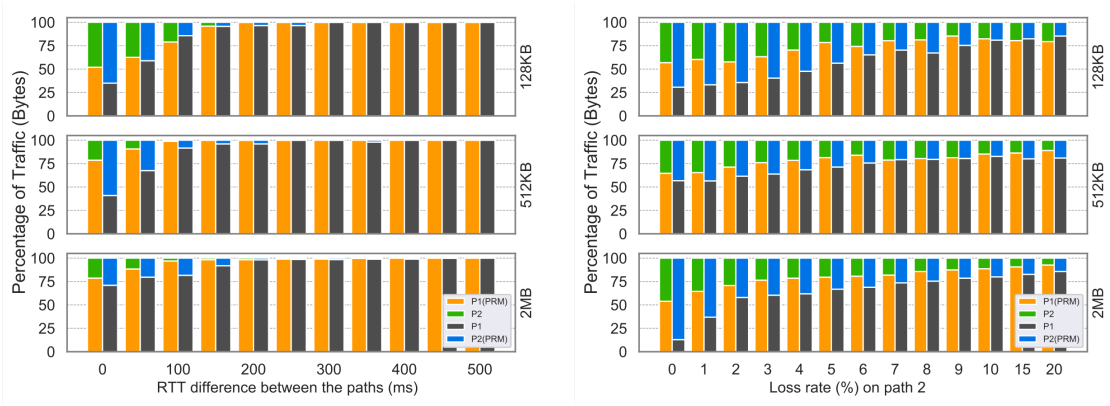
6.5.2 Methodology

In this section, we examine the performance of the default MPTCP scheduler for simple web downloads using `iPerf`. According to [162], about 58% of today's global Internet traffic is attributed to video streaming. Consequently, content streaming is typically preceded by (for instance, metadata files) and accompanied by (change in user preferences) small download sessions that are characteristically short flows. This suggests that a bottleneck in such download sessions could result in poor quality of experience (QoE) at the user end. Therefore, we again explore the download performances of 128KB, 256KB, 512KB, 1MB, and 2MB file sizes using `iPerf`. Similar to Section 6.4, we vary the latency or loss rate on path 2, keeping the other parameter steady. Inter-path latency is varied from 50ms through 500ms, with increments of 50ms. Loss rates are varied from 1% through 10%, including 15% and 20%. Furthermore, we manually specify the primary path at the start of each experiment to study the effect on performance when a higher RTT or more lossy path is initially selected. To determine the fraction of Internet traffic carried over each subpath, we capture network traces at the server-side using `tcpdump`.

Then, we investigate the distribution of the page load times for Tranco top 1K websites [36]. For evaluation purposes, we cache the top 1000 webpages on our server located in Virginia, along with their associated web objects. We then establish an MPTCP connection from the client to the server. We run the experiment 100 times for each webpage and calculate the average page load time while replicating the path characteristics in Section 6.4, i.e., we vary the latency and loss rates on path 2.

6.5.3 Evaluation

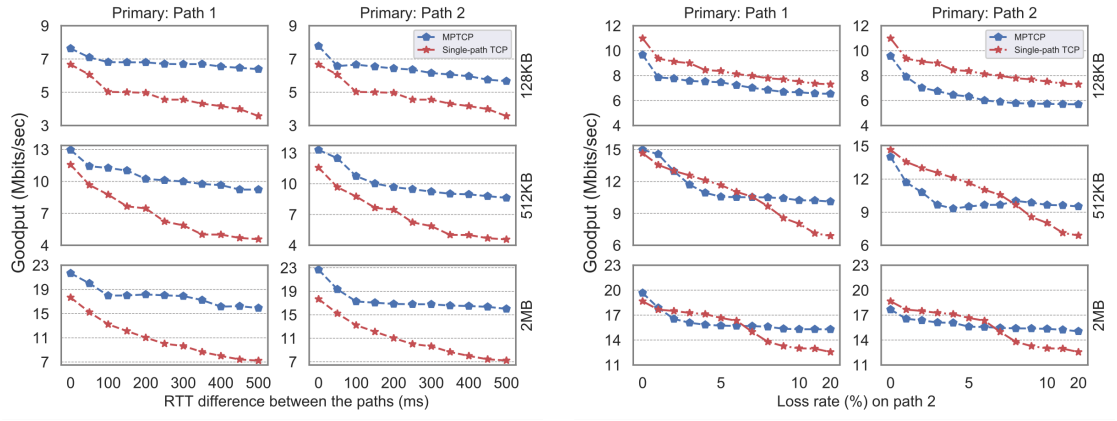
In this section, we present the results of our real-world experiments. We first study varying path characteristics' effect on short flows' web downloads. Figure 6.8 shows the percentage of traffic sent over the two subpaths as the path characteristics (latency and loss rate) change. We observe in Figure 6.8(a) that traffic on path 2 becomes negligible as inter-path latency increases. An expected behavior can be attributed to lower RTT on path 1, which is the single greatest deciding factor in the default scheduler. On the other hand, while loss rates increase on path 2, selecting a viable path becomes less deterministic, as we hypothesized. We see a similar trend in Figures 6.8(b) and 6.4(b), which indicates that subflows fail to exit their slow-start phase before the download is complete, resulting in considerable traffic over the lossier path. However, the implications of that behavior become more noticeable in our study of goodput.



(a) RTT on path 2 increases (b) Loss rate on path 2 increases

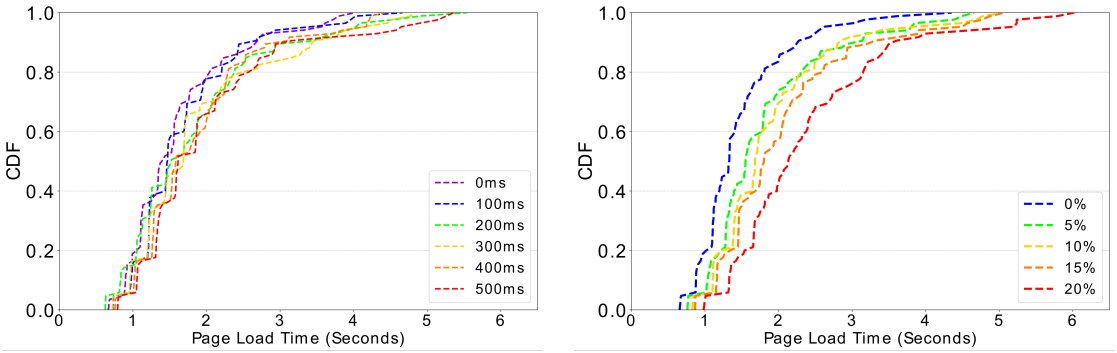
Figure 6.8: Fraction of traffic carried on each path for real world iPerf experiments.

Figure 6.9 illustrates the goodput for 128KB, 512KB, and 2MB file sizes as the path properties become more heterogeneous. In Figure 6.9(a), we notice that the difference in goodput between MPTCP and single-path TCP widens as the inter-path



(a) RTT on path 2 increases (b) Loss rate on path 2 increases

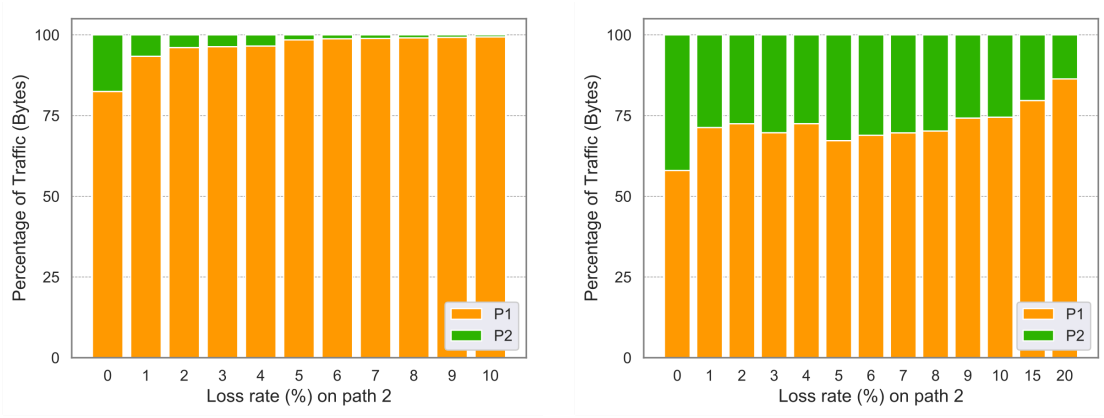
Figure 6.9: Achieved goodput for web downloads (higher is better).



(a) RTT on path 2 increases (b) Loss rate on path 2 increases

Figure 6.10: CDF of page load time (lower is better).

RTT increases. This is supported by Figure 6.8(a), which demonstrates that after an inter-path latency difference of 150ms, traffic is almost exclusively carried on the lower latency path 1, enabling it to achieve better goodput than the slower single-path TCP. In contrast, Figure 6.9(b) shows that single-path TCP indeed surpasses MPTCP with uneven lossy paths, which confirms our findings from the in-lab experiments (Figure 6.5(b)). For instance, in the case of the 128KB file, MPTCP almost always performs worse than single-path TCP. Single-path TCP achieves better goodput for 512KB and 2MB files until 6% and 8% loss rates, respectively. This is a significant finding since it is less likely for a path to have loss rates as high as 6%–8% under nor-



(a) RTT on path 2 increases (b) Loss rate on path 2 increases

Figure 6.11: Fraction of web traffic carried on each path for real world experiments.

mal operating conditions. Stated otherwise, single-path TCP almost always performs better than MPTCP while downloading short flows. In practice, this could mean the difference between an instant video playback versus laggy, pause-filled video content. This result informs us of how crucial the initial path selection process is, which can adversely affect the end-user experience.

Another essential factor that affects user experience is the webpage load time. As shown in Figure 6.10(a), the detrimental effect of increasing RTT on path 2 is subtly absorbed by MPTCP since path 1 carries the bulk of the traffic (Figure 6.11(a)). Instead, page load time is adversely affected when loss rates are introduced on path 2. Given the path selection criteria for the default MPTCP scheduler, which only considers minimum RTT, it is not surprising to observe that more than 25% of the total traffic is carried on path 2, even though it suffers from a significant 10% packet loss. Correspondingly, there is a steady increase in page load time in Figure 6.11(b) as path 2 becomes lossier. Since cellular networks tend to be lossier than Wi-Fi [141], the path selection process becomes highly critical. Web surfing and video streaming are everyday activities for an average mobile user and could easily experience a decline in QoE. The results shown here indicate the limitations of the default scheduler and

the factors that need to be incorporated to improve mobile performance.

6.6 Discussion and Conclusion

The results of our measurement study point to a few important findings:

Round-trip Times. Our results in Section 6.3 show that round-trip times on cellular are substantially higher than on Wi-Fi. In addition, the majority of web servers resolved on cellular have lower round trip times on Wi-Fi than on cellular.

Reachability. The primary interface is responsible for DNS resolution in MPTCP. Resolving IPs over cellular can be detrimental since many servers are unreachable through Wi-Fi or cellular. MPTCP clearly cannot be used in those scenarios. For MPTCP to succeed, CDNs deployed by commercial providers may need to be modified to be reachable from the outside (i.e., not behind NAT). However, this proposition contrasts with how content is delivered on the modern Internet. These observations hint at the low viability of a full-scale MPTCP deployment.

Heterogeneous Paths. Our measurement study shows that diversity in path loss rates is ignored when selecting the best path. This weakness is embedded inside the default scheduler because it considers only RTT as a metric for path selection. We realize that it is difficult to implement MPTCP with heterogeneous subpaths. From an institutional level, near-homogeneous network conditions are needed on all subpaths for MPTCP to utilize its full potential. For MPTCP to achieve optimal performance, it should take a broader view of path performance and, at a minimum, consider loss rate.

Part II

System Design and Predictive Analysis

Chapter 7

Inferring Network Performance

7.1 Current Landscape

The domain of computer networks has evolved tremendously over the past 50 years. This evolution has loosely followed a typical pattern: characterize current networks, and address the inefficiencies through new systems and standards. Frequently, the design philosophy of those systems is reactive, i.e., when a degradation in performance is observed, the system kicks into action. In our mission to connect the next billion users to the Internet, we argue that we need to focus closely on not just proactive but predictive systems. The reasons why there is a need for predictive systems are:

1. Access to usable Internet is critical. Mobile broadband, an effective access technology achieved through commercial Long-Term Evolution (LTE) and 5G cellular networks, provides a meaningful way of expanding this access [18]. Still, it is often concentrated in urban areas, leaving economically marginalized and sparsely populated regions underserved [6]. The US Federal Communications Commission (FCC) incentivizes cellular operators serving rural areas [29, 30]

and maintains transparency by releasing maps from each operator showing geographic areas of coverage [46]. Recently third parties have challenged the veracity of these maps, claiming these maps over-represent actual coverage and thus may discourage much-needed investments. Most of these claims, however, are either focused on limited areas where a few dedicated researchers can collect controlled coverage measurements (e.g., through wardriving) or are mainly qualitative in nature [31, 10, 32]. As dependence on mobile broadband connectivity increases, especially in the face of the COVID-19 pandemic, predictive systems that quantitatively validate FCC coverage datasets at scale are becoming acutely necessary to evaluate and direct resources in Internet access deployment efforts [83, 127]. This is an issue of technology and technology policy, with equity and fairness implications for society.

2. As growing numbers of people depend on cellular networks as their primary means for accessing healthcare, financial, and educational services, and it has become *critical to evaluate how well these networks service their end users*. For instance, the COVID-19 pandemic has increased reliance on mobile and portable Internet access. As a result, communities without access to usable, high-speed broadband, such as many rural and tribal regions, are severely disadvantaged [26]. While many characterization studies exist in literature [10, 98, 51, 117], there is a need to evaluate and infer how well the deployed network performs over time. Moreover, these assessments need to be scalable and easy to deploy. The Federal Communications Commission has encouraged researchers to study and report the state of rural networks [85] – the design and deployment of a system that can report network performance can address this goal.

3. Few prior studies have focused on assessing mobile broadband in rural areas of

the US; there is a lack of accessible datasets that are comprehensive (including network-level and application-level traces) and representative and inclusive of rural demographics. Unfortunately, the evaluation of user quality of experience (QoE) for commonly used applications accessed over cellular in regions where people are most likely to be smartphone dependent [45, 24, 6] poses a significant scalability challenge. QoE metrics collection over mobile broadband in a geographic area requires time and resource-intensive measurements for each network provider. As a result, experiments at a single geographic point can be lengthy. Moreover, in rural areas, obtaining mobile Internet measurements in places where people are likely to use mobile broadband (e.g., at their homes or along local transportation corridors) can be challenging [35], as places of interest are far apart (requiring more resource-intensive targeted measurement campaigns) and less densely populated (prohibiting representative crowd-sourcing measurement efforts). As a result, disruptive systems that can infer the QoE for commonly used applications over mobile broadband at scale are urgently needed.

7.2 Why Now?

Today, more than ever, we need predictive systems for the following reasons:

1. Recent years have seen a massive infusion of funds from governmental agencies globally to upgrade the broadband (mobile and fixed) infrastructure. For instance, The Broadband Equity, Access, and Deployment (BEAD) Program provides $42.45 billion to expand high-speed internet access by funding planning, infrastructure deployment, and adoption programs in all 50 US states [163].

Similarly, BharatNet is a project envisioned by the government of India to connect all the villages of India digitally [164]. The European Union supports the Connectivity for a European Gigabit Society program through fund allocation on a 5-year basis [165]. However, allocating funds to specific sectors and planning zones commensurate with the extent of the upgrade is challenging. Given the difficulty in aggregating precise network performance and the expected user demand in the coming years, public sector establishments base their allocation on heuristics and reports from (not completely unbiased) service providers. Since these massive funds can unlock access to usable Internet for billions of people, policymakers require granular datapoints on current and anticipated network demand. Scalable, predictive systems can help achieve that goal.

2. Assessment of continuously increasing (or decreasing) user demand is imperative for infrastructure planning. Access to systems that can predict user demand on a macro scale is potentially groundbreaking. This is because effective radio network planning addresses coverage and capacity requirements and, at the same time, enables network expansion without the need for significant changes at existing sites. With the introduction of 5G mmWave, where deployments are typically denser and frequent upgrades are challenging, systems with the ability to predict precise end-user experience elevate the probability of a well-provisioned network.

3. While the knowledge of user demand as it relates to capacity planning on a macro scale is essential, so is the observability and data about sudden increases in demand. Over-utilization of an under-provisioned network, typical during large gatherings (e.g., street festivals, protests) and post-disaster scenarios, can lead to a partial or complete failure of the communication system [13]. Tradi-

tionally, cellular networks are akin to walled gardens where observability into the network is limited at best. This problem can be alleviated using systems that can assess network state using minimally invasive methods, preferably with passive measurements.

7.3 Solution

This thesis takes a step towards addressing the critical need for predictive systems. In Chapter 8 through Chapter 11, we develop several systems capable of inferring the network performance from the providers and the end users' perspective. Our design philosophy is centered around three elements: (i) robustness, (ii) scalability, and (iii) ease of deployment.

1. In Chapter 8, we propose a novel system, Lumos, that can infer overload conditions in packet-switched cellular networks (LTE/5G) using only passive measurements. Lumos enables observability into cellular networks without the cooperation of service providers.

2. Chapter 9 extends our work to develop Edain, a comprehensive network monitoring suite that works in conjunction with Lumos to infer network congestion in LTE and 5G networks accurately.

3. Chapter 10 introduces several state-of-the-art network models that can predict end-user QoE with a high degree of accuracy using inexpensive, readily available network measurements.

4. In Chapter 11, we present AKIDA, a platform that leverages network accelerators (i.e., SmartNICs) to enable in-network processing of transient workloads.

AKIDA is designed to be an autonomous compute platform with auto-scaling capabilities.

7.4 Challenges and Limitations

This thesis examines predictive systems that are designed to be robust and scalable. These systems are some of the first studies to analyze network performance without the infeasibility in deployment. Consequently, there are certain challenges and limitations associated with first-of-its-kind studies:

1. Building a predictive system using supervised learning requires the collection of tens of thousands of data points to tune network models. Unfortunately, few to no existing datasets focus on QoE measurements at scale and around networks that exhibit wide temporal and spatial variation. As a result, we collect, clean, and curate all of our datasets used in training the models described in this thesis. Undertaking measurement campaigns is intensive in terms of human effort, economic cost, and time, making it difficult to scale. We outlined the specific challenges in Chapter 2 (§2.2).

2. Certain systems require an auxiliary training of a small batch of crowd-sourced data. For instance, database entries are crowd-sourced to collect cellular signal measurements when the recording device passes within range of an eNodeB. This dataset is more comprehensive in population centers and highways where people more often travel. If there are too few measurements overall, or if measurements are primarily sourced from the same grid section, then the eNodeB location estimate can be inaccurate. In addition, certain coverage data required for training are generated using predictive models that are proprietary to the

operator [9] and not generally reproducible. Furthermore, the publicly available dataset consists of binary coverage and lacks performance-related data.

3. Accurate prediction of network events (like degradation in QoE performance) involves collecting datasets that include those adverse events. Such datasets comprise rare adverse events, as most of the data points are normal, i.e., they do not have very many QoE degrading events. As a result, those datasets have the class-imbalance problem, typical for most anomaly detection problems. To address this issue, we apply the sampling technique SMOTE [166] to balance the classes artificially. However, such an approach reduces the number of data points we can use for training, say, the classifier, which affects the accuracy. With SMOTE, we observed no improvements in accuracy with simpler learning models (e.g., SVM, random forest, etc.) and lower accuracy for more complex, neural network-based classifiers. Therefore, for the NN-based classifiers, we adopt a new technique that has been proven to increase classification accuracy in datasets that suffer from the class-imbalance issue for the object detection problem [167]. This technique addresses the class-imbalance problem by re-shaping the standard cross-entropy loss so that it lowers the weights for the majority class [167]. It also introduces the concept of *focal loss* that prevents the majority class from overwhelming the classifier during the training phase.

Our goal with this body of work is to set the direction of networking research towards predictive systems that are reproducible, verifiable, open-sourced network models. We hope to encourage other researchers and the academic community to develop and deploy open-sourced systems.

Chapter 8

Packet-level Overload Estimation in LTE Networks using Passive Measurements

8.1 Introduction

With 6 billion users and growing, LTE is the leading mobile network technology worldwide in 2022 [168]. With this growth comes critical challenges in sustaining consistent, high-quality service to an increasing subscriber base [169]. In a well-provisioned region, sudden escalation in traffic demand from user equipment (UEs) can occur during large gatherings (e.g., street festivals, protests). Similarly, damaged infrastructure and atypical utilization volume can overwhelm a previously well-provisioned network after a disaster. Prior work has also demonstrated that even in areas that cellular providers claim are well-covered, persistent over-usage due to insufficient capacity can exist [14].

As a specific example, in 2017, Hurricane Maria brought down 95% of cellular

sites in Puerto Rico [13]. As a result, affected citizens on the ground were unable to request a rescue from rising flood waters. In disaster scenarios, call volume may overload capacity even when cellular towers remain functional, causing base stations to reject calls [170, 171]. Unfortunately, cellular providers are incentivized to state that damaged cellular services have been returned to an operational state. Indeed, after Hurricane Maria, statuspr.org soon reported that over 90% of cell towers were operational; however, anecdotal evidence indicated such statistics were grossly overstated.

To remedy this disparity between reported coverage and actual usability, individual users, watchdog groups, and government agencies need tools to verify whether a network serves customers adequately. After a disaster, the FCC typically receives outage reports from telecoms, for instance [11]. Still, due in part to overload, the actual usability of active towers is difficult to assess without access to the internal network. Public entities should be able to determine a particular base station's overload and operational status/usability. Further, they should be able to accomplish this without relying on the cooperation of a cellular provider.

To address this critical need, we propose a novel solution to infer overload in LTE networks based on messages broadcast by the eNodeB. Through the analysis of multiple message types, we draw clear comparisons between instances of high network utilization and typical operating conditions for several eNodeBs. Our results indicate that eNodeBs demonstrate measurable performance differences indicative of overload conditions.

Importantly, our solution works without the cooperation of the cellular provider. Using low-cost, readily usable off-the-shelf equipment, we demonstrate that unencrypted broadcast messages sent by the eNodeB [172] on the broadcast channel can be passively collected and analyzed to estimate local overload and hence usability.

We quantify our results by computing two normalized metrics, which are proportional to the number of connection reject messages and cell barring signals (`cellBarred`), respectively (cell barring signals prohibit UEs from camping on a particular cell). In addition, we evaluate the back-off timer (`waitTime`) encapsulated in each reject message. In LTE, a connection reject message does not contain a rejection case. Consequently, we must use higher `waitTime` values, coupled with increased rates of connection request denials, to reveal possible overload.

To test the operation of our system, we perform multiple measurement campaigns: two at events with unusually large crowd gatherings and two at those exact locations but during typical usage. We collect and analyze over 3.2 million LTE frames through these measurement campaigns. Our analysis indicates that overload on an eNodeB can be identified through an increase in reject messages and mean back-off time. Moreover, these events are often accompanied by a significant increase in cell barring signals. We show that overloaded cell towers frequently deny larger percentages of connection requests and issue higher `waitTime` compared to typical utilization periods. Further, we observe an unusual number of barring signals prohibiting UEs from camping on their desired eNodeBs.

8.2 Related Work

Diagnostic methods in LTE networks are known to be cumbersome. This includes packet-level analysis to estimate overload because messages transmitted after the connection establishment stage are invisible to a passive device. As a result, little prior work leverages passive measurements to detect overload.

Previous work has led to the development of several network analysis tools. `xgoldmon` [173], for instance, can monitor control plane messages over 2G/3G but

not LTE. `SCAT` [174] is a tool designed to detect problems in cellular networks, which, although quite useful, is limited to only active monitoring on Qualcomm and Samsung basebands. `QXDM` [175] is a tool developed to diagnose network statistics that is limited to only Qualcomm baseband and requires a paid license. While [176, 177, 178] offer very similar feature sets to the tools discussed above, they are not tailored to work with software-defined radios for passive monitoring. Schmitt et al. [179, 14] employ a comparable approach to ours, except their study is limited to GSM networks. We believe the biggest drawback of these prior tools is their inability to work with passive measurement devices, such as software-defined radios (SDRs).

Several prior works have studied various congestion control algorithms in LTE networks [180, 181, 182, 183], but little work has explored overload detection without involving an active monitoring aspect. Torres et al. [184] use machine learning models to predict network congestion. However, their approach requires considerable historical data. It is not suitable for urban sectors where eNodeBs are upgraded regularly to cater to increasing user bases, nor can it be used to assess current overload levels. Chakraborty et al. [185] introduce LoadSense, which offers a measure of cellular load using channel sensing at the PHY layer. Similarly, [186] allows a client to monitor the LTE base station's PHY-layer resource allocation efficiently and then map such information to estimate available bandwidth. Cellular Link Aware Web loading (CLAW) is proposed in [187], which boosts mobile Web loading using a physical-layer informed transport protocol. Although the aforementioned tools can estimate whether the radio resources are fully allocated, they do not explicitly reveal whether the network is overloaded.

Our method focuses primarily on analyzing messages broadcast *before* a connection is established, as these messages can be captured and analyzed by low-cost SDRs. Our approach is portable, scalable, independent of any proprietary platform

(e.g., Qualcomm, Samsung, etc.), and works with any cellular service.

8.3 Background

In our work, we examine cellular transmissions using software-defined radios. While most of the transmissions on LTE are encrypted between the eNodeB (LTE base station) and UE (user equipment, such as a cellphone) [188], connection establishment messages are sent in the clear. We use these messages to determine overload, as described in the following sections.

8.3.1 Radio Resource Control (RRC)

The RRC protocol [189, 190] supports the transfer of common Non-Access Stratum (NAS) [191] information (which applies to all UEs) as well as dedicated NAS information (which applies only to a specific UE). Directed RRC messages (unicast to a single UE) are transferred across Signalling Radio Bearers (SRB)s, which are mapped onto logical channels [192, 193] – either the Common Control CHannel (CCCH) during connection establishment or a Dedicated Control CHannel (DCCH) if the UE is in an active connection state. Similarly, System Information (SI) messages are mapped to the Broadcast Control CHannel (BCCH). Since messages on DCCH are on a private channel, they cannot be decoded by passive monitoring devices.

Common Control CHannel (CCCH): This channel is used to deliver control information in uplink and downlink directions when there is no confirmed association between a UE and the eNodeB – i.e.,,, during connection establishment. Messages on this channel are transmitted in the clear and can be passively decoded. We leverage this knowledge to analyze signaling messages and estimate the overload level in an

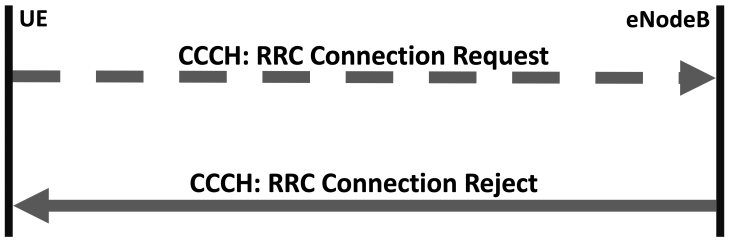


Figure 8.1: Flow diagram for connection reject message.

eNodeB.

Broadcast Control CHannel (BCCH): This is a downlink channel that is used to broadcast System Information (SI). It consists of the Master Information Block (MIB) and a number of System Information Blocks (SIBs). The MIB and SIBs are broadcast through Radio Resource Control (RRC) messages. SIB1 is carried by `SystemInformationBlockType1` message. Though there are other SI messages, we focus on SIB1 for this study. SIB1 contains the cell barring (`cellBarred`) status, which indicates whether or not a UE may choose the cell. When `cellBarred` status is indicated, the UE is not permitted to select/reselect this cell, not even for emergency calls [194]. In that case, the UE may connect to another cell.

8.3.2 Signalling Radio Bearers

A Signalling Radio Bearer (SRB) [195] carries CCCH signaling data. An SRB is used during connection establishment to establish the Radio Access Bearers (RABs) and deliver signaling while on the connection (for instance, to perform a handover, reconfiguration, or release). There are three types of SRBs. SRB0 uses the CCCH channel with *transparent mode RLC* while SRB1 and SRB2 use the dedicated channel with *acknowledged mode RLC*. In other words, SRB0 can be decoded by non-network equipment such as a software-defined radio in the vicinity, while SRB1 and SRB2

cannot. Table 8.1 shows various signalling messages SRB0 carries.

For our study, we focus on `RRCConnectionReject` messages (solid arrow in Figure 8.1) with corresponding `waitTime` (back-off time, before a UE can again initiate a connection) values, `Connectio- nRequest` messages, and `cellBarred` signals (BCCH). We formulate two normalized metrics based on the percentage of reject messages per request sent and the ratio of `cellBarred` signals to the number of SIB1 messages transmitted over thirty-second time bins.

8.3.3 Managing Overload

Overload management is invoked to unburden a cell to an acceptable level when overload is detected, for instance, if the cell load remains above a threshold for some continuous period. An alternative strategy, such as that used by WCDMA, is to lower the bit rates of connected users until the load returns to an acceptable level [196]. However, in a pure packet-based system such as LTE, the user bit rate is maintained at the MAC scheduler [197], which already provides a soft degradation of user throughput as the system load increases. Thus, if overload is detected in a cell, the system must remove a subset of the connected bearers until the load is reduced to an acceptable level. Admission Control [198] is used to restrict the number of UEs given access to the system to provide acceptable QoS to admitted users.

8.4 Implementation

8.4.1 Experimental Setup

In our experimental setup, our receiver is comprised of an Ettus Research USRP B210 [199] SDR attached to an MPantenna SUPER-M ULTRA Mobile Antenna with

Table 8.1: SRB0 Summary

Channel Type	**RLC Mode**
CCCH	Transparent (*Decodable from passive capture*)
Direction	**RRC Message**
Downlink	RRC Connection Setup RRC Connection Reject
Uplink	RRC Connection Request

a frequency range from 25MHz to 6GHz [200]. The USRP is connected to a Lenovo ThinkPad W550s laptop for data collection and post-processing. We use the `srsUE` mode in the open-source srsLTE software suite [81] to locate available cells in the vicinity by scanning all frequency bands. On the day of the event, we capture broadcast messages in the form of binary I/Q samples using srsLTE's UE `usrp_capture` utility.

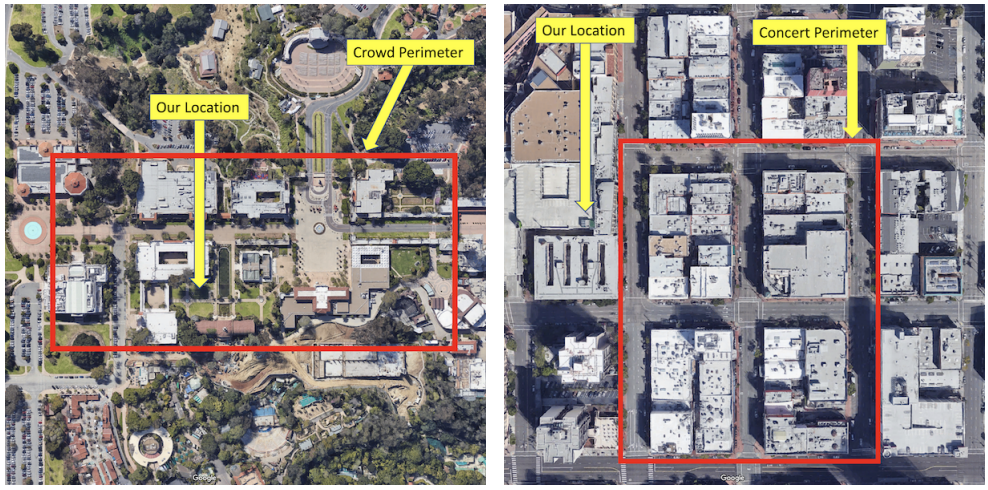
8.4.2 LTE Packet Decoding

We start with converting binary I/Q samples to hexdumps. To investigate the extent of overload on eNodeBs, we transform the hexdump into network traces using Wireshark's *text2pcap* command [201]. Next, we use *lte_rrc* Lua dissectors to decode LTE RRC messages using *tshark* [202]. We employ $lte-rrc.dl.ccch$ and $lte-rrc.ul.ccch$ protocols to decode RRC messages on the downlink and uplink common control channel, respectively. Lastly, we use the $lte-rrc.bcch.dl.sch$ protocol to decode downlink messages on the broadcast control channel.

Listing 8.1 shows a snapshot of the decoded RRC message on the downlink CCCH. We can see the *RRCConnectionReject* message tree and additional options sent by the eNodeB during the RRC connection establishment phase. Embedded in this message is the *waitTime* parameter. While reject messages indicate overload,

Listing 8.1: Snapshot of a decoded DL - CCCH message showing RRCConectionReject.


```
"user_dlt": "DLT: 147, Payload: lte-rrc.dl.ccch \
 (LTE Radio Resource Control (RRC) protocol)",
"lte-rrc.DL_CCCH_Message_element": {
 "per.choice_index": "0",
 "lte-rrc.message": "0",
 "lte-rrc.message_tree": {
  "per.choice_index": "2",
  "lte-rrc.c1": "2",
  "lte-rrc.c1_tree": {
   "lte-rrc.rrcConnectionReject_element": {
    "per.choice_index": "0",
    "lte-rrc.criticalExtensions": "0",
    "lte-rrc.criticalExtensions_tree": {
     "per.choice_index": "0",
     "lte-rrc.c1": "0",
     "lte-rrc.c1_tree": {
      "lte-rrc.rrcConnectionReject_r8_element": {
       "per.optional_field_bit": "1",
       "lte-rrc.waitTime": "6"
       "lte-rrc.nonCriticalExtension_element": {
        "per.optional_field_bit": "1",
        "per.optional_field_bit": "1",
        "per.octet_string_length": "2048",
        "lte-rrc.lateNonCriticalExtension":
         "34:07:79:f0:2c:e7:90:00:28:07:63:48:31:b7:90:00:
          38:07:04:f0:22:67:81:08:30:87:9e:40:3f:37:60:70:
          20:27:82:00:21:17:4c:88:36:47:80:00:20:07:15:00:
          2a:97:90:00:28:17:95:30:2a:97:99:30:2c:87:82:00:
          21:07:4c:f0:36:77:85:b0:22:d7:82:30:21:07:82:40:
          21:27:9f:80:2f:d7:68:18:33:f7:84:00:32:07:23:80:
          21:d7:76:f0:2b:77:91:40:28:a7:81:00:30:97:42:00:
          21:17:88:70:24:27:96:00:2b:07:48:00:24:17:66:00:
          23:d7:93:c0:29:f7:94:00:3a:07:50:f0:38:77:68:80:
```



((a)) **SPD dataset**.
Balboa Park, San Diego, CA

((b)) **CSR dataset**.
Downtown San Diego, CA

Figure 8.2: Google aerial map of experimental datasets.

we can use the value of the *waitTime* metric to measure the severity of overload. The value of *waitTime* is an integer in the range of 1–16, which defines how many seconds the UE should wait after reception of the *RRCConnectionReject* until a new connection can be attempted. According to 3GPP TS 23.401 [203], when rejecting an RRC connection request, the eNodeB indicates to the UE an appropriate timer value that limits further requests relative to the severity of overload; the more the overload, the greater the *waitTime*. Upon receiving the *RRCConnectionReject* message, the UE starts timer T302 [189], with the timer value set to *waitTime*. The UE cannot send another *RRCConnectionRequest* for mobile originating calls, signaling, terminating access, or circuit-switched fallback (CSFB) [204, 205] on the same cell until the expiry of T302. Note that in LTE, the *RRCConnectionReject* message does not contain a *RejectionCause* therefore, *waitTime*, in conjunction with reject messages, is a crucial parameter in assessing the level of overload.

8.4.3 Datasets

To test our proposed solution, we identify times and locations in which we anticipate cellular overload, capture traces, and then compare network performance in those traces with baselines captured in the exact location during normal operating conditions (when no network overload is likely to occur). We select spaces that are anticipated to have large gatherings but are unlikely to be provisioned for large crowds (i.e., city streets as opposed to stadiums that typically have sufficient network capacity to handle crowds).

We hypothesize that during large crowds, we will observe higher numbers of `RRCConnectionReject` messages than in times of regular operation. Overall, our dataset consists of over 3.2 million frames, with data collection that lasts for a cumulative duration of about 5.2 hours. While it is not possible to compute the exact number of UEs in the vicinity due to the lack of international mobile subscriber identity (IMSI) number in broadcast messages for security reasons, measuring the number of temporary unique UE IDs (`uniqueUeID`) in RRC Connection Requests allows us to estimate the number of active UEs present nearby. A detailed overview of the datasets is provided in Chapter 2 (§2.1)(St. Patrick's Day, San Diego).

8.5 Evaluation

We analyze five RRC elements: (a) `RRCConnectionReject`, (b) `wait- Time`, (c) `RRCConnectionRequest`, (d) `cellBarred` signal and (e) number of SIB1s transmitted (#SIB1). Collectively, we refer to this data as "RRC metrics". We plot the values of these RRC metrics over thirty-second bins. We found that thirty-second bins were appropriate for our analysis because smaller time bins had little to no relative varia-

tion between the samples; however, we missed important data points when we used sixty-second or larger bins. In our evaluation, we observe that the rate of transmitted `RRCConnectionReject` messages is considerably higher in SPD and CSR than in their respective baselines, per our initial hypothesis. Further, we discover an increase in *cellBarred* signals and *waitTime* values in overloaded datasets (i.e., SPD and CPR).

8.5.1 Rejects

According to [195], an eNodeB may send an `RRCConnectionReject` in response to the UE's `RRCConnectionRequest` for exactly one of the following three reasons: (i) the eNodeB is overloaded (e.g., severe increase in requesting UEs that the eNodeB cannot accommodate); (ii) the necessary radio resources for the connection setup cannot be provided (for instance, damaged equipment on eNodeB that results in limited access to the core network); or (iii) the Mobility Management Entity (MME) is overloaded. The MME is the key control node for the LTE access network, which serves several eNodeBs. It is in charge of all the control plane functions related to subscriber and session management. Once the MME detects overload, it transmits an `overload start` message to the affected eNodeBs, signaling them to reject connection request messages for non-emergency and non-high priority mobile devices originated services.

Analysis of the reject messages sent over a fixed time interval can quantify the level of overload in the network. Figure 8.3 illustrates the average number of reject messages transmitted in thirty-second bins. As predicted, we notice significantly more reject messages in the SPD and CSR datasets. Figure 8.3(a) indicates that there are, on average, eight times more reject messages during the SPD parade compared to the SPD baseline (Figure 8.3(b)). Similarly, we observe a fifteen-fold increase in reject

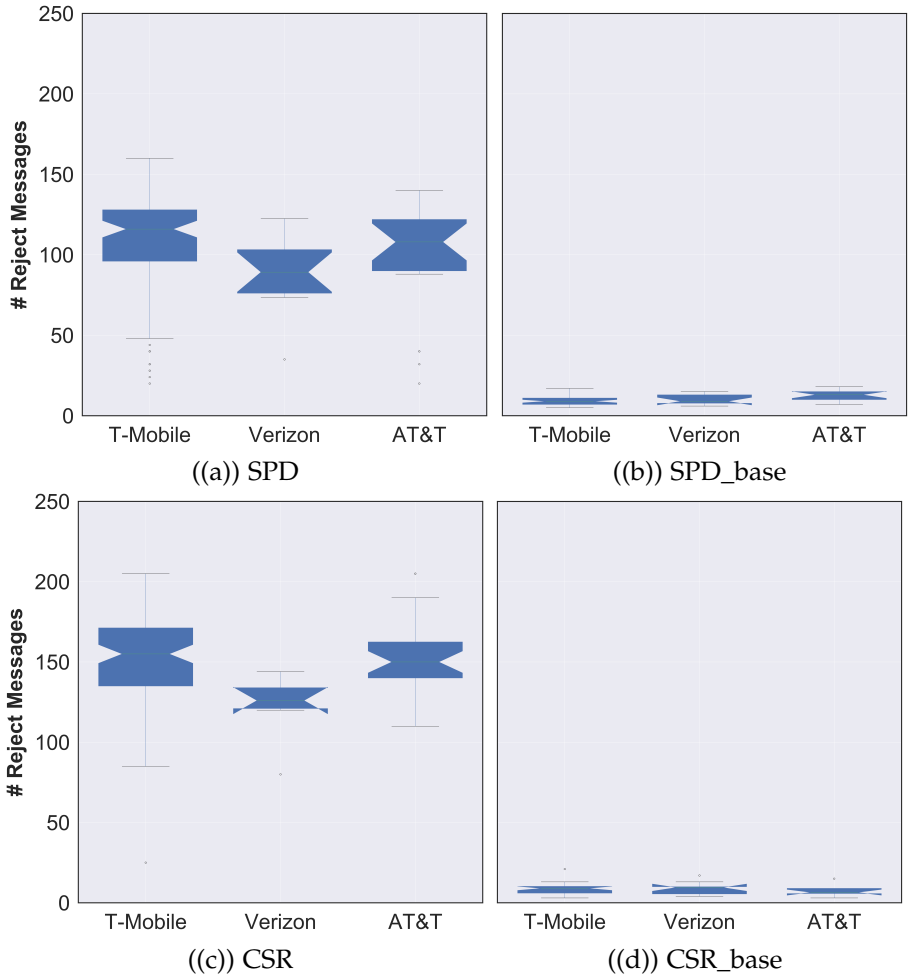


Figure 8.3: Number of `RRCConnectionReject` messages transmitted in thirty-second bins.

messages in Figure 8.3(c) as compared to Figure 8.3(d). This significant increase in reject messages indicates an increase in cellular network utilization.

8.5.2 Phi (Φ) Measure

We examine a normalized second-order metric to understand better how overload levels vary. We define the Phi (Φ) measure as the ratio of the number of `RRCConnectionReject` messages to the number of `RRCConnectionRequest` messages. Once again, we choose a bin size of 30 seconds. The Phi measure indicates the sever-

ity of overload, as it reflects the percentage of new users who could not connect to the network. In future studies, we wish to examine the temporal variation in Phi (or the number of new users that were rejected) to quantify the maximum acceptable load threshold in eNodeBs. As expected, there is a considerable difference between overloaded datasets (i.e., SPG and CSR) and their respective baselines. Figure 8.4(a) shows that Phi is as much as three times that in Figure 8.4(b). This difference is even more pronounced in Figure 8.4(c), where Phi is more than seven times that in Figure 8.4(d). This trend is similar to what we observed in Section 8.5.1. It also indicates the relationship between the number of UEs (# `uniqueUeIDs`) and the tendency towards network overload, as is expected.

8.5.3 Average `waitTime`

When we compare the average `waitTime` across datasets in Figure 8.5, we observe that SPD and CSR have longer `waitTimes` than their baselines. We also see that AT&T performs worse in SPD, closely followed by T-Mobile. In CSR, T-Mobile appears to perform slightly worse than AT&T. Verizon, however, shows lower `waitTime` in all of the datasets. Note that the sample sizes of these distributions are proportional to the number of reject messages, as shown in Figure 8.3. Nevertheless, all telecom providers transmit longer `waitTimes` during increases in traffic demand.

Longer `waitTime` in SPD and CSR is perhaps explained by the high proportion of UEs (# `uniqueUeIDs`) in the given location. Suppose the magnitude of UEs is great enough to result in overload. In that case, eNodeBs start curtailing overload conditions by engaging proprietary mitigation schemes, one of which is transmitting longer `waitTime`. The overall result confirms our hypothesis that these messages and parameter values can be used to infer overload. The comparison is noteworthy as

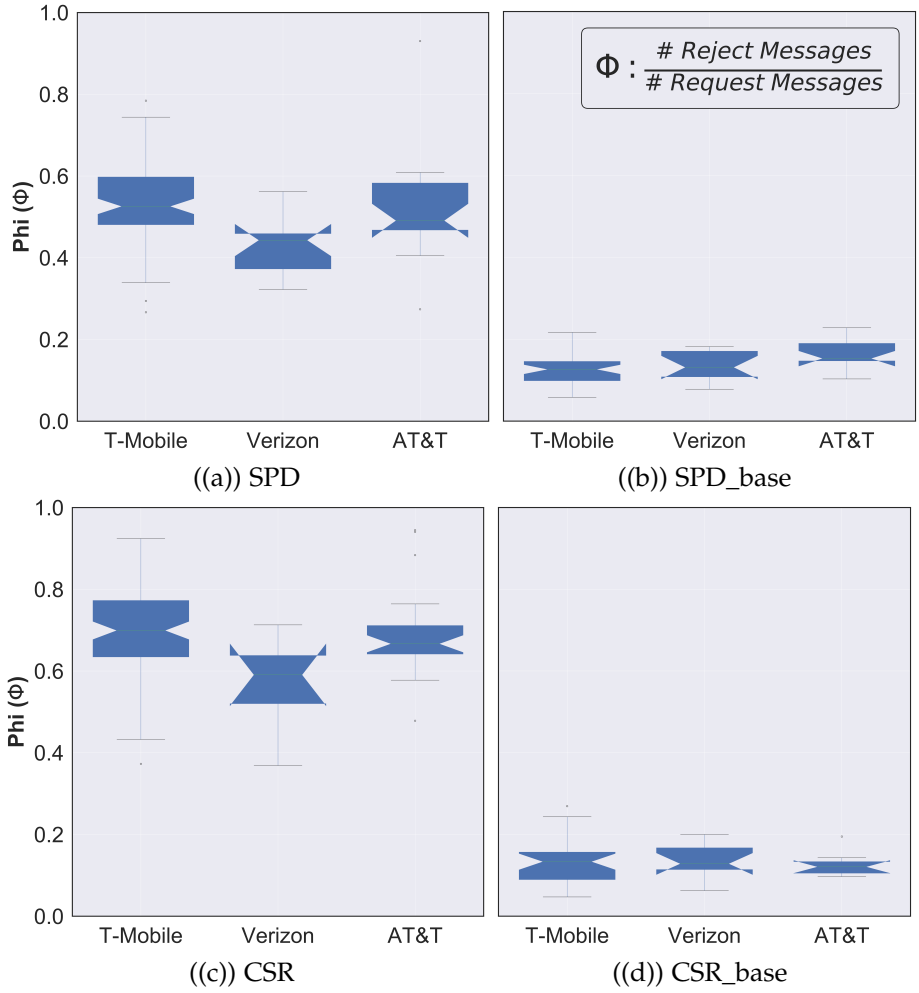


Figure 8.4: Phi (Φ) measure in thirty-second bins.

it supports our earlier results where we compute `RRCConnectionReject` messages. Average `waitTime` serves as an additional indicator of overloaded eNodeBs.

8.5.4 Omega (Ω) Measure

In addition to the reject messages and their corresponding `waitTime`, `cellBarred` status is a crucial parameter that can indicate overload in an eNodeB. The `cellBarred` status transmitted within a system information block 1 (SIB1) message indicates that the UE is not allowed to camp on a particular cell. We suspect that cells can initiate

load balancing during overload conditions by systematically preventing UEs from anchoring on them. To evaluate our theory, we analyze `cellBarred` messages to compare the percentage of these messages in our datasets.

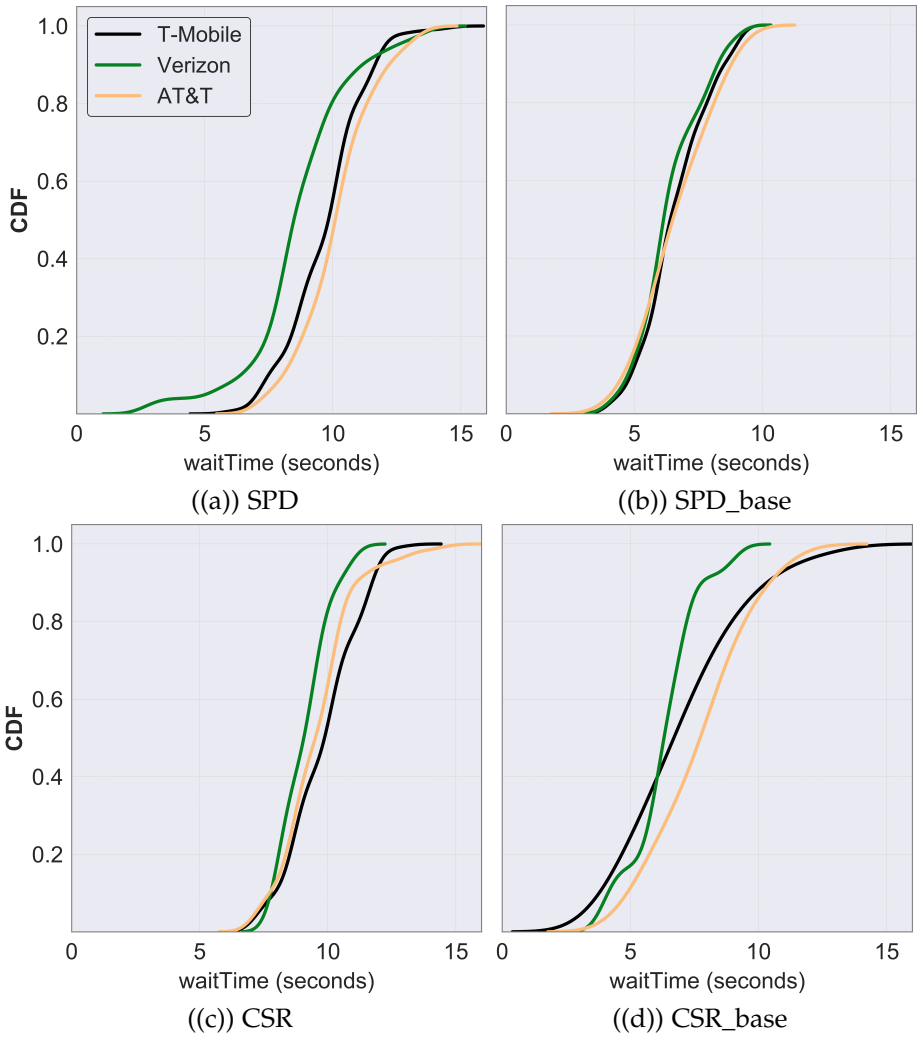


Figure 8.5: Distribution of average `waitTime`.

The Omega (Ω) metric allows us to measure the ratio of *cellBar– red* signals transmitted to the number of SIB1 frames received in thirty-second bins. We use this second-order metric to establish a correlation between Omega and overload. Figure 8.6 depicts the variation in Omega across all datasets. We observe an increase of

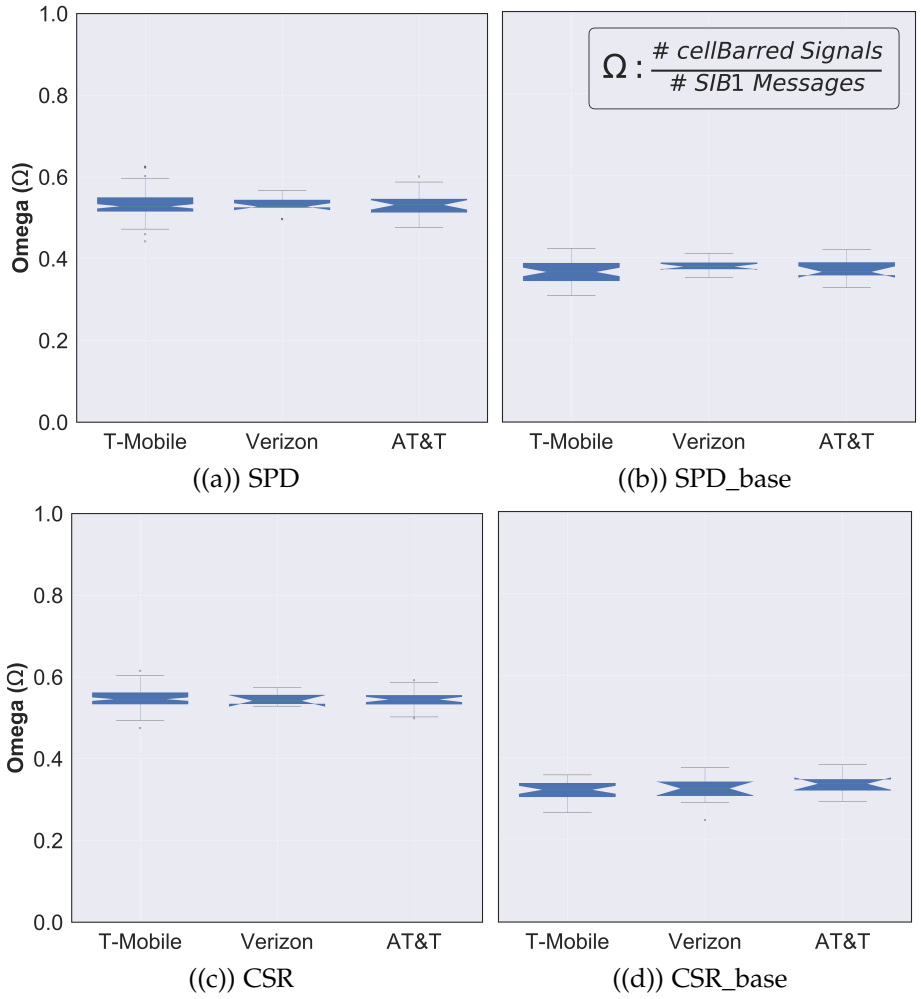


Figure 8.6: Omega (Ω) measure in thirty-second bins.

20% in SPD and CSR datasets over their respective baselines. This indicates a relationship between cell barring signals and overload, confirming our hypothesis. However, it is interesting to observe that each of the mobile network operators (i.e., T-Mobile, Verizon, and AT&T) have comparable Omega values in SPD and CSR, even though they exhibit noticeably different trends in Figures 8.3 and 8.4. That is because the inherent load-handling capacity of eNodeBs and the density of users served to differ. This suggests that overloaded eNodeBs operating in disparate network conditions prefer consistently rejecting incoming connection requests rather than broadcast un-

availability (through cell barring messages), regardless of their proprietary overload mitigation schemes.

8.6 Conclusion

In this chapter, we propose a novel method to assess overload in nearby LTE eNodeBs, utilizing off-the-shelf hardware without requiring cooperation from the cellular provider. Our analysis offers convincing evidence that messages broadcast by the eNodeB can be used to detect cellular overload using passive monitoring. In future work, we will explore how passive overload inference can be leveraged in a system for automated overload mapping using ground-based data collection and Unmanned Aircraft Systems, independent of collaboration from a cellular provider. Such tools can be leveraged by regulators and policymakers and allow targeted deployment of alternative communication channels. In the next chapter, we explore how this novel method can be extended to infer network congestion and the performance degradation observed on UEs.

Chapter 9

Estimation of Congestion from Cellular Walled Gardens using Passive Measurements

9.1 Introduction

In this study, we propose a novel solution[1] to infer overload and congestion[2] in LTE networks based on messages broadcast by the eNodeB. We develop Lumos, a data analysis platform capable of quantifying overload in eNodeBs. The design and implementation of Lumos are described in section 9.2. To validate the existence of congestion as detected by Lumos, we develop a network monitoring suite that au-

[1]This study is an extension of Chapter 8 where the focus was on estimating overload. In this study, we build upon our prior work to detect and quantify congestion as a result of the estimated overload on the network.

[2]We consider overload as the state where the user equipment is denied access to camp on an LTE base station (eNodeB) due to the current number of connections. In contrast, congestion is a state while a user device is connected, leading to performance degradation at the user end (e.g., slower downloads, poor video streaming quality, etc.). We explain these two terms in more detail in section §9.2.1.

tomates the collection of Quality of Service (QoS) and Quality of Experience (QoE) metrics; this suite is described in section 9.3. Through the analysis of multiple message types, we draw clear comparisons between instances of high network utilization and typical operating conditions for several eNodeBs. Further, we evaluate performance differences incurred from overload-driven congestion through QoS and QoE metrics assessment. Our results indicate that eNodeBs demonstrate measurable performance differences indicative of overload conditions and network congestion.

Importantly, our solution works without the cooperation of the cellular provider. Using low-cost, readily usable off-the-shelf equipment, we demonstrate that unencrypted broadcast messages sent by the eNodeB on the broadcast channel can be passively collected and analyzed to estimate local overload. We concurrently collect measurements on active monitoring devices to draw parallels between overload, network congestion, and network usability.

We quantify our results by computing two normalized metrics, which are proportional to the number of connection reject messages and cell barring signals (`cellBarred`), respectively (cell barring signals prohibit UEs from camping on a particular cell). In addition, we evaluate the back-off timer (`waitTime`) encapsulated in each reject message. Note that in LTE, a connection reject message does not contain a rejection cause. Consequently, we must use higher `waitTime` values, coupled with high rates of connection request denials, to indicate possible overload. To validate our results, we use our network monitoring suite (§9.3) to demonstrate the corresponding performance degradation at the network (QoS) and user-level (QoE). For instance, under high load, QoE for common applications such as Web browsing and video streaming can deteriorate to the point of unusability.

To test the operation of our system, we perform multiple measurement cam-

paigns[3]: three at events with unusually large crowd gatherings, and three at those exact locations but during times of typical usage. Through these measurement campaigns, we collect and analyze over 7 million LTE frames from four major telecom operators in the US (AT&T, Sprint, T-Mobile, and Verizon).

Our key contributions and findings include:

- Overload on an eNodeB can be identified through an increase in reject messages and mean back-off time. We show that overloaded cell towers frequently deny 4× larger percentages of connection requests, issue 35% higher `waitTimes`, and broadcast unavailability through 30% more barring signals than baseline measurements;

- Overload conditions are often accompanied by a significant increase in congestion as revealed through a considerable drop in service usability at the user end. We observe at least 10× lower throughput, 2× higher latencies, and 8× higher packet losses in atypical utilization periods;

- Quality of experience significantly drops for video streaming applications: we note a minimum of 6× higher start-up delay, 3× lower video quality, 3× higher stall ratio, and over 30% decrease in buffer levels.

9.2 Lumos: Detecting Overload

To examine and quantify cell load on eNodeBs, we develop Lumos. Lumos is based on the idea that third-party assessment tools should be accessible to the community and carry a low hardware footprint. Our design philosophy is driven by implementing comprehensible systems that are easy to understand and orchestrate.

[3]We plan to release the dataset used for this study publicly.

9.2.1 Congestion Control

While overload can potentially impair services at the eNodeBs, congestion can lead to severe performance degradation at the end-user. Congestion refers to the performance bottleneck experienced by users as a result of significantly higher traffic demand by the UEs. Overload is the scenario that causes the network to deny UEs to preserve the load balancing capabilities of the eNodeBs. An overload scenario will manifest as a depletion of resources critical to the network's operation.

Congestion control is invoked to unburden a cell to an acceptable level when overload is detected, for instance, if the cell load remains above a threshold for some continuous period. An alternative strategy, such as that used by WCDMA, is to lower the bit rates of connected users until the load returns to an acceptable level [196]. However, in a pure packet-based system such as LTE, the user bit rate is maintained at the MAC scheduler [197], which already provides a soft degradation of user throughput as the system load increases. Thus, if congestion is detected in a cell, the system must remove a subset of the connected bearers until the load is reduced to an acceptable level. Admission Control [197] is used to restrict the number of UEs given access to the system to provide acceptable QoS to admitted users.

9.2.2 Design

We orchestrate a passive end-user system capable of listening to all broadcast messages over the air, a functionality available in all UEs. In our setup, the receiver comprises an Ettus Research USRP B210 [199] SDR attached to an Apex III Wideband 5G/4G Dipole Terminal Antenna with a frequency range from 450MHz to 6GHz [80]. The USRP is connected to a Lenovo ThinkPad W550s laptop for data collection, post-processing, and analysis. We greatly reduce computational overheads on the laptop

by implementing our custom Lua dissectors to decode and parse the LTE packets. We use the `srsUE` mode in the open-source srsLTE software suite [81] to locate available cells in the vicinity by scanning all frequency bands. On the day of the event, we capture broadcast messages in the form of binary I/Q samples using srsLTE's UE `usrp_capture` utility. Figure 9.1 shows our experimental setup. LTE packet decoding is instrumented similar to the previous study. The details are provided in Chapter 8 (§8.4.2).

9.2.3 Applicability in 5G Cellular Networks

This section discusses the scalability of this methodology to 5G cellular networks. After a thorough comparison of 5G NR RRC protocols in 3GPP TS 38.331 [206], we find that the signaling procedure for NAS is identical to that of LTE [189]. Furthermore, *RRCConnectionRequest*, *RRCConnectionReject* and *waitTime* retain the same message body extensions as LTE. We also observe LTE-equivalent protocols to acquire MIB and SIB1 in 5G NR RRC, with similar periodic intervals between each message broadcast by the base station (referred to as gNodeB in 5G). We believe Lumos can be scaled to be deployed on 5G networks with little to no modification in the source code and hardware requirements for base stations using the sub-6GHz

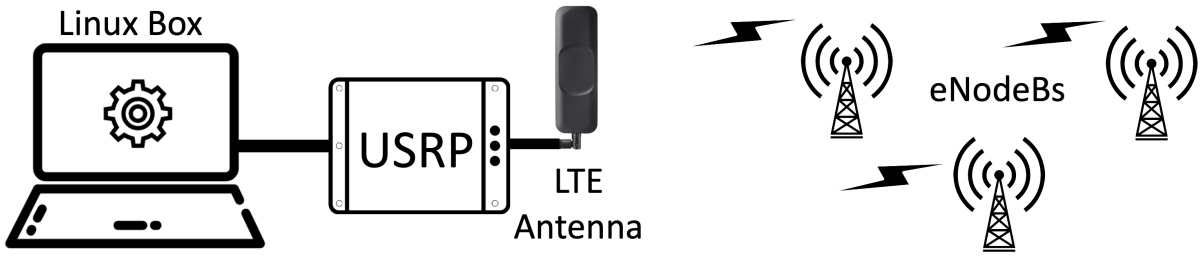


Figure 9.1: Experimental setup.

spectrum. For mmWave deployments (>30 GHz), we need a compatible USRP and radio antenna to implement Lumos, with no changes to the source code. Due to the limited availability of 5G coverage and 5G-enabled phones, we could not gather 5G specific data at the time of our measurement collection.

9.3 Network Monitoring Suite

We develop a comprehensive network monitoring suite to quantify congestion due to cellular overload. Whereas Lumos is a passive monitoring platform, this monitoring suite is used for active network measurement. The suite provides an extensive set of features to measure QoS and QoE metrics at the client. We have used this active measurement tool in sixteen locations across the United States to compare mobile broadband performance under varying network conditions.

9.3.1 Implementation

The monitoring suite's functionality ranges from computing network level (throughput, latency, and packet traces) to application-level (on-demand video streaming (YouTube)) and page load time measurements. We measure cellular performance by tethering phones to laptops running the monitoring suite. We ensure that the cellular plans on all our devices have unlimited data and are hot-spot enabled to effectively achieve the same level of performance as we would on the mobile device. This tool was developed for Linux, keeping ease of deployment in mind. It is agnostic to the network type and provides flexible deployment in wired, Wi-Fi, or cellular environments. Developing an integrated smartphone app was impractical as the level of unification achieved for various application-level measurements (YouTube, Skype,

etc.) was simply not possible on smartphone operating systems, given the walled access to the iOS ecosystem and recent restrictions introduced in Android APIs [207].

Latency: The monitoring suite's `rtt_out` function automates the collection of round-trip times by initiating pings through Hping3 [96] to a server hosted on an AWS instance (Virginia). We configure Hping3 to use TCP packets instead of ICMP. The ping duration is capped at 120 seconds, with one-second intervals between each ping. The average latency is then computed using two different sessions - one before the throughput tests (described below) and one after. This enables us to capture the latency variation introduced in the network, if any, after a throughput measurement session. We observe an average round-trip time of 61ms with a standard deviation of $\pm$3 ms across all baseline measurements.

Throughput: To calculate the achieved throughput, we initiate iPerf threads to download a 10 MB file from the same AWS instance as the latency test. The measurement is repeated 10 times, and results are saved on the client-side. We concurrently collect packet traces at the client to compute second-order metrics such as packet loss.

Page Load Time: Load times are initiated through the `plt_stream` function. We automate the loading of Web pages using Selenium [105]. For our measurements, we use the Tranco Top 25 list [36]. To evaluate load times, we log the performance timings of a Web page starting from *navigationStart* through the *loadEventEnd* event. These instances of event timings support fine-grained analysis of page load times. We set the monitoring suite to run `plt_stream` three times to estimate load times better. The browser cache is automatically wiped after each Web page load to reflect the true load time for the next iteration.

Video Streaming (YouTube): Examination of QoE metrics from on-demand video streaming services is a challenging problem, mainly because of encrypted traffic, as

demonstrated by prior work [208, 209]. Because of the wide proliferation of video applications, user experience for streaming services is critical on mobile broadband. We built the *video_stream* function into the monitoring suite to log QoE metrics from YouTube videos. To execute this experiment, we first automate the loading and playback of the YouTube video on the Chrome browser using Selenium [105]. The video resolution is set to auto. Then we use YouTube's iframe API [108] to capture playback events reported by the video player. The API outputs a set of values that indicate player state (not started, paused, playing, completed, buffering) using the getPlayerState() function. The API also provides functions for accessing information about playtime and the remaining buffer size. To ensure uniformity across all our datasets, we loop a 180-second video for every location and cellular operator three times.

9.3.2 Datasets

We identify times and locations in which we anticipate cellular overload (§9.4.1), capture traces, and then compare network performance in those traces with baselines captured in the same location during normal operating conditions (when no network overload is likely to occur). We select spaces that are anticipated to have large gatherings but are unlikely to be provisioned for large crowds (i.e., city streets as opposed to stadiums, which typically have sufficient network capacity to handle crowds).

We hypothesize that during large crowds, we will observe higher numbers of `RRCConnectionReject` messages than in times of regular operation. We demonstrate in [90] that first- and second-order metrics derived from `RRCConnectionReject` messages can assess overload in nearby LTE eNodeBs. In this study, we collect three new extensive datasets from several locations across California. Further, to establish the effect of overload on network congestion and user experience, we undertake

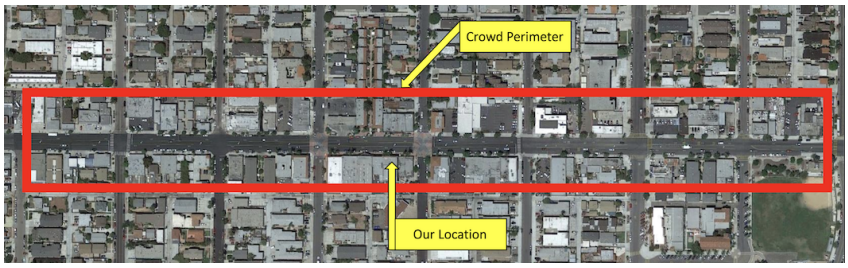
Table 9.1: Dataset Information

Locations	Duration	# LTE Frames	# `uniqueUeID`
ADM	129 mins	1.63M	59,084
CWF	126 mins	1.89M	67,728
AIS	149 mins	2.34M	111,404
ADM Base	57 mins	316K	5,307
CWF Base	62 mins	442K	7,478
AIS Base	65 mins	396K	6,089

a measurement study to synchronously aggregate QoS and QoE metrics. Overall, our dataset consists of over 7 million LTE frames for overload estimation, with data collection that lasts for a cumulative duration of about 10 hours. While it is not possible to compute the exact number of UEs in the vicinity due to the lack of international mobile subscriber identity (IMSI) number in broadcast messages for security reasons, measuring the number of temporary unique UE IDs (`uniqueUeID`) in RRC Connection Requests allows us to estimate the number of active UEs present nearby. We concurrently collect network level (throughput, latency, and packet loss) and application-level (YouTube streaming and page load times) measurements using the monitoring suite. To avoid reiterations, the monitoring suite is run alongside Lumos for all the datasets described below. Table 9.1 provides an overview of the datasets. A detailed overview of the datasets is provided in Chapter 2 (§2.1)(Santa Fe, New Mexico; Suburbs, San Diego; San Francisco).

9.4 Evaluation

We begin our analysis by studying the broadcast messages transmitted by eNodeBs. We observe that the transmission rate for `RRCConnectionReject` messages can accurately indicate the network overload state. Further, we evaluate QoS and



((a)) **ADM dataset**. Adams Avenue, San Diego, CA

((b)) **CWF dataset**. Waterfront Park, San Diego, CA

((c)) **AIS dataset**. Palace of Fine Arts, San Francisco, CA

Figure 9.2: Google aerial map of experimental datasets.

QoE metrics to learn that severe congestion is introduced during network overload, leading to user experience degradation.

9.4.1 Overload Analysis

We analyze five RRC elements: (a) `RRCConnectionReject`, (b) `waitTime`, (c) `RRCConnectionRequest`, (d) `cellBarred` signal and (e) number of SIB1s transmitted (#SIB1). Collectively, we refer to this data as "RRC metrics." We plot the values of these RRC metrics over thirty-second bins. We find that thirty-second bins are appropriate for our analysis because smaller time bins have little to no relative variation between the samples; however, we miss important data points when we use sixty-second or larger bins. Our evaluation indicates that the rate of transmit-

ted `RRCConnectionReject` messages in all locations is at least 4× higher than their respective baselines, per our initial hypothesis. Further, we discover an increase of more than 30% in *cellBarred* signals and 35% higher *waitTime* values in overloaded datasets (i.e., ADM, CWF, and AIS). For all of the following results, plots are color-coded corresponding to their respective operator's logo for readers' convenience and easy understanding.

Rejects

According to [195], an eNodeB may send an `RRCConnectionReject` in response to the UE's `RRCConnectionRequest` for exactly one of the following three reasons: (i) the eNodeB is overloaded (e.g., severe increase in requesting UEs that the eNodeB cannot accommodate); (ii) the necessary radio resources for the connection setup cannot be provided (for instance, damaged equipment on eNodeB that results in limited access to the core network); or (iii) the Mobility Management Entity (MME) is overloaded. The MME is the key control node for the LTE access network, which serves several eNodeBs. It is in charge of all the control plane functions related to subscriber and session management. Once the MME detects overload, it transmits an `overload start` message to the affected eNodeBs, signaling them to reject connection request messages for non-emergency and non-high priority mobile devices originated services.

Analysis of the reject messages sent over a fixed time interval can quantify the level of overload in the network. Figure 9.3 illustrates the average number of reject messages transmitted in thirty-second bins. As predicted, we notice significantly more reject messages in the overloaded datasets (ADM, CWF, and AIS). Figure 9.3(a) indicates that, on average, Sprint and T-Mobile networks broadcast eight times more reject messages during ADM as compared to the ADM baseline (Figure 9.3(b)). We

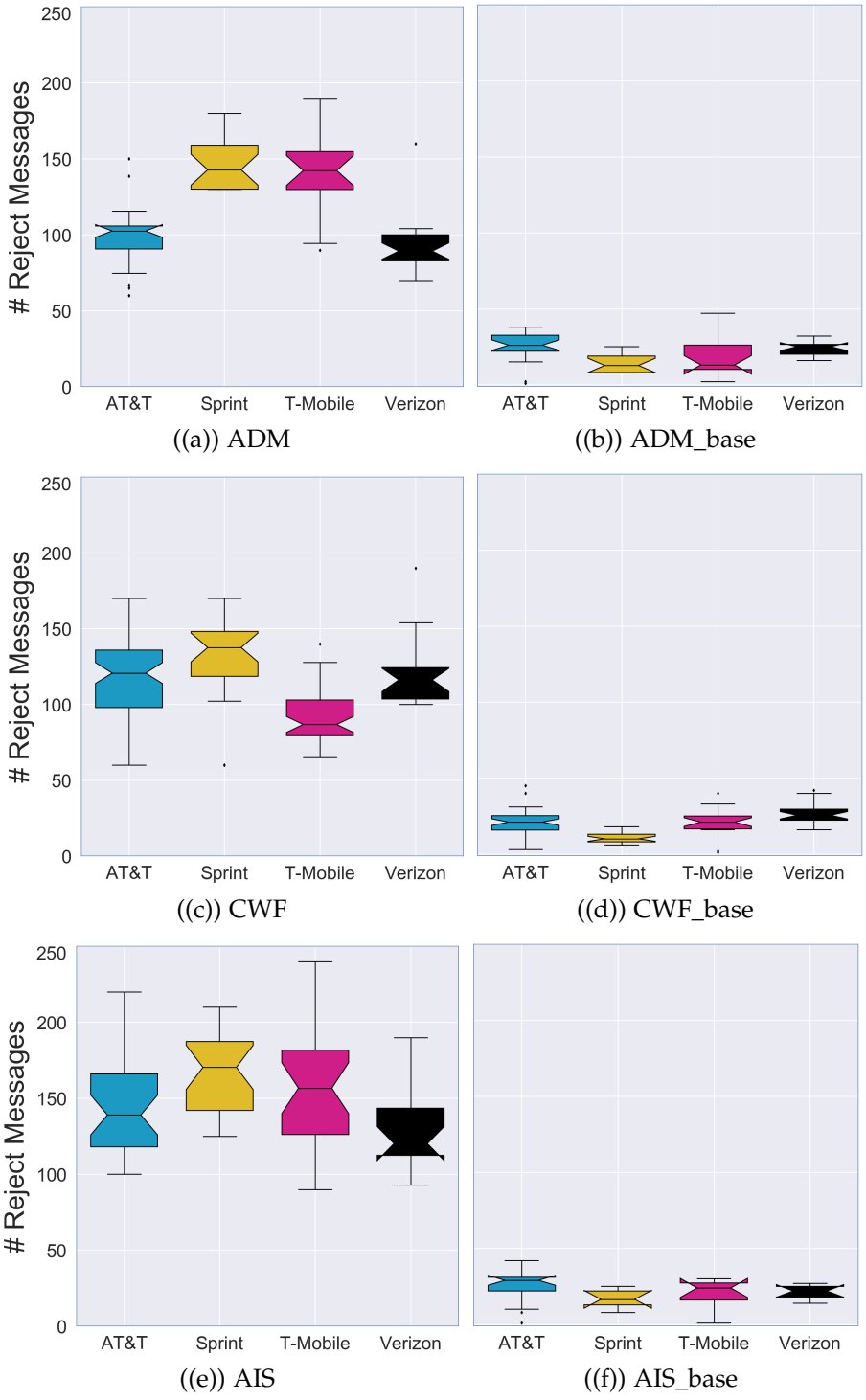


Figure 9.3: Number of `RRCConnectionRejectmessages` transmitted in thirty-second bins.

see that AT&T and Verizon are slightly less overloaded with about a 4× increase from their respective baselines. In CWF, we observe a similar (∼5×) increase in rejects for all networks except T-Mobile, which reports a marginally lower increase (as shown in Figure 9.3(c)). Finally, Figure 9.3(e) (AIS) displays considerably more reject messages for all operators, as compared to their respective baselines in Figure 9.3(f). We posit that this trend is due to the dense presence of participants, as demonstrated by over 2 million captured LTE frames. Upon closer inspection of the AIS dataset, we detect 5.5×, 9×, 7× and 6× increase in `RRCConnectionRejectmessages` for AT&T, Sprint, T-Mobile and Verizon, respectively. The significant increase in reject messages clearly indicates an upsurge in network utilization.

Phi (Φ) Measure

We examine a normalized second-order metric to understand better how overload levels vary. We define the Phi (Φ) measure as the ratio of the number of `RRCConnectionReject` messages to the number of `RRCConnectionRequest` messages. Once again, we choose a bin size of 30 seconds. The Phi measure indicates the severity of overload, as it reflects the percentage of new users who could not connect to the network. In future studies, we plan to examine the temporal variation in Phi (or the number of new users that were rejected) to quantify the maximum acceptable load threshold in eNodeBs. The overall trend is similar to what we observed in Section 9.4.1. It also indicates the relationship between the number of UEs (# `uniqueUeIDs`) to the tendency towards network overload, as is expected.

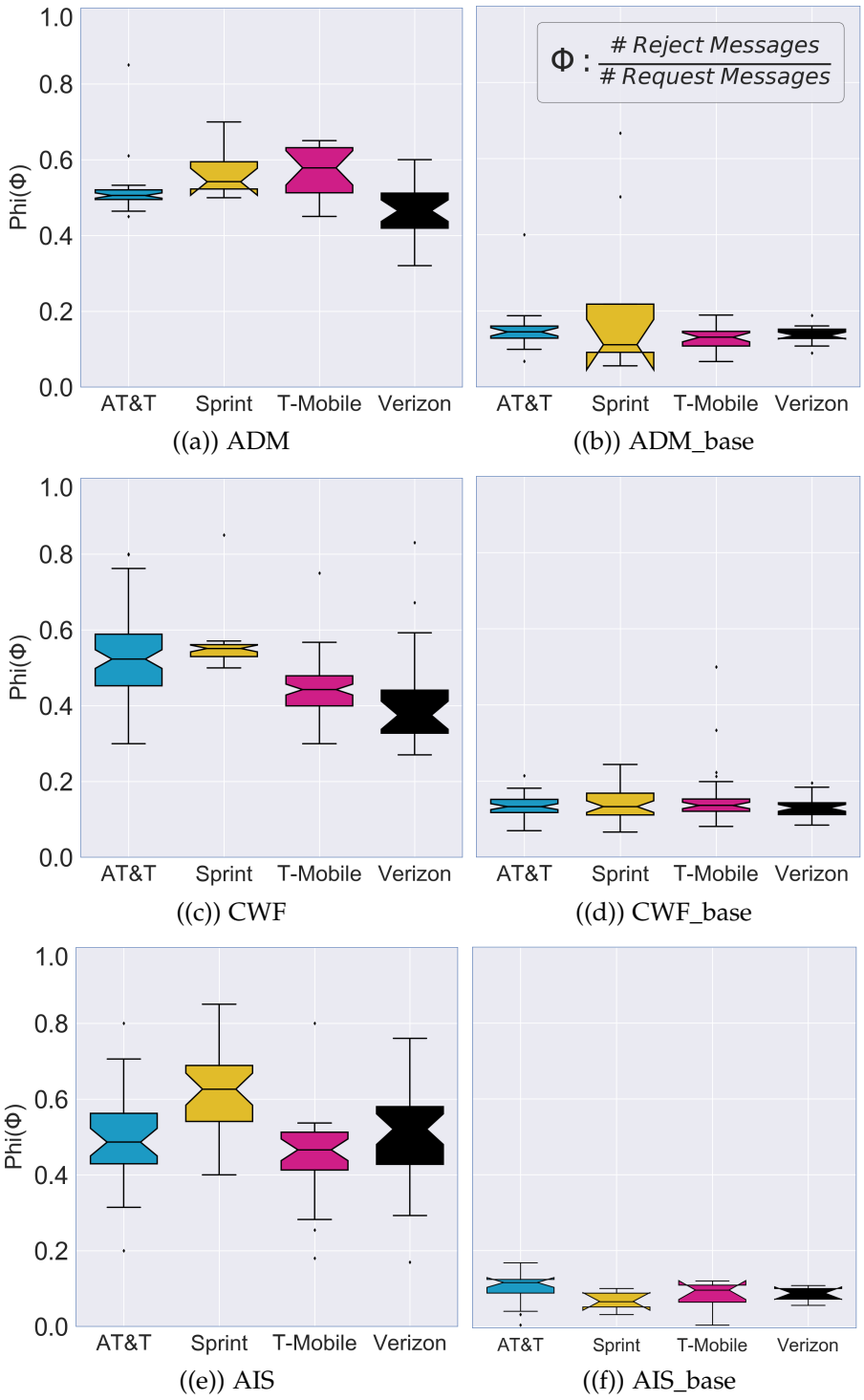
Our examination reveals a remarkable difference between overloaded datasets (i.e., ADM, CWF, and AIS) and their respective baselines. Figure 9.4(a) shows that Phi is more than three times that in Figure 9.4(b). This difference is even more pronounced in the CWF dataset. Figure 9.4(c) shows an increase of about 5.5×, 5.5×,

4.5× and 3.5× in Phi measure, respectively, as compared to Figure 9.4(d). Sprint under-performs in our evaluation of Phi, with over 8× difference observed between AIS and AIS baseline. Further, we also note a considerable variance in Sprint at ADM baseline. This result suggests that Sprint's infrastructure at Adams Avenue is under-provisioned for normal operating conditions compared to other networks. Overall, Sprint's network appears to have the least ability to handle a sudden escalation in user demand.

Average `waitTime`

When we compare the average `waitTime` across datasets in Figure 9.5, we observe that overloaded datasets on the left have longer `waitTimes` than their baselines. An exception occurs in ADM, where Sprint produces lower `waitTimes` during baseline measurements. This is likely indicative of lower loads, which would confirm our previous hypothesis that Sprint has far fewer subscribers (at least in ADM) and yet is still under-provisioned for local events such as the Adams Fair. Surprisingly, Verizon shows modestly higher `waitTime` in CWF despite reporting relatively lower Phi levels in Figure 9.4(c). In AIS, Sprint performs slightly worse than others, with an average deviation of ∼4 seconds from its baseline. Note that the sample sizes of these distributions are proportional to the number of reject messages, as shown in Figure 9.3. Nevertheless, all telecom providers transmit longer `waitTimes` during increases in traffic demand.

Longer `waitTime` in ADM, CWF, and AIS is perhaps explained by the high proportion of UEs (# `uniqueUeIDs`) in the given location. Suppose the magnitude of UEs is great enough to result in overload. In that case, eNodeBs start curtailing overload conditions by engaging proprietary mitigation schemes, one of which is transmitting longer `waitTime`. The overall result confirms our hypothesis that these messages and



((a)) ADM ((b)) ADM_base

((c)) CWF ((d)) CWF_base

((e)) AIS ((f)) AIS_base

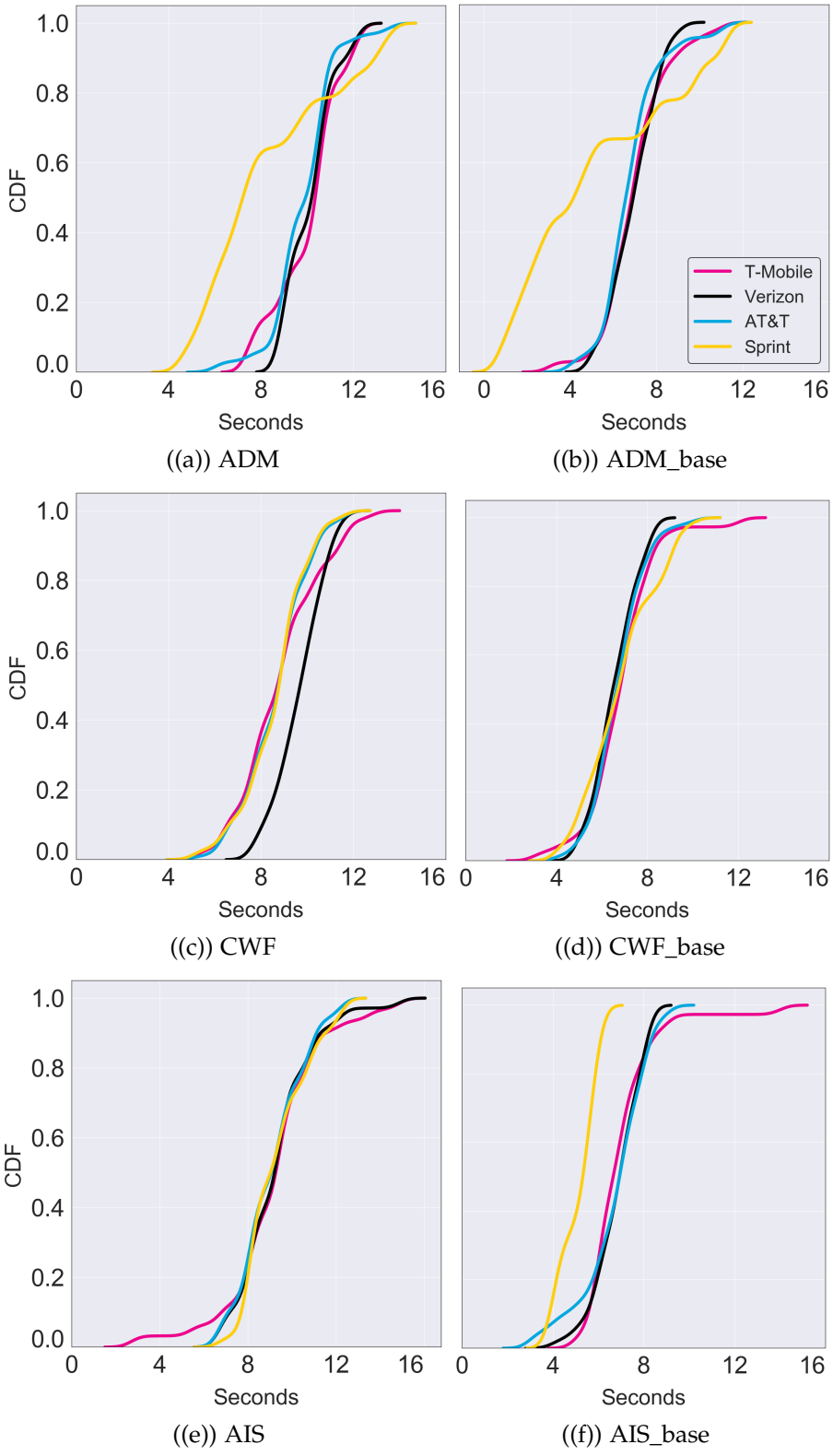
Figure 9.4: Phi (Φ) measure in thirty-second bins.

parameter values can be used to detect overload. The comparison supports our earlier results where we compute `RRCConnectionReject` messages. Average `waitTime` serves as an additional indicator of overloaded eNodeBs.

Omega (Ω) Measure

In addition to the reject messages and their corresponding `waitTime`, `cellBarred` status is another parameter that can indicate overload in an eNodeB. The `cellBarred` status transmitted within a system information block 1 (SIB1) message indicates that the UE is not allowed to camp on a particular cell. We suspect that cells can initiate load balancing during overload conditions by systematically preventing UEs from anchoring on them. To evaluate our theory, we analyze `cellBarred` messages to compare the percentage of these messages in our datasets.

The Omega (Ω) metric allows us to measure the ratio of *cellBarred* signals transmitted to the number of SIB1 frames received in thirty-second bins. We use this second-order metric to establish a correlation between Omega and overload. Figure 9.6 depicts the variation in Omega across all datasets. We observe an increase of 30% in ADM and CWF datasets over their respective baselines and about a 45% increase in AIS. This indicates a relationship between cell barring signals and overload, confirming our hypothesis. However, it is interesting to observe that each of the mobile network operators has comparable Omega values in the overloaded datasets, even though they exhibit noticeably different trends in Figures 9.3 and 9.4. This is similar to the trend we discovered in [90]. Moreover, the AIS baseline has a 15% decrease in Omega measure compared to other baselines. That is because the inherent load-handling capacity of eNodeBs and the density of users served apparently differ. This suggests that overloaded eNodeBs, even those that operate under different network conditions, prefer to consistently reject incoming connection requests



((a)) ADM ((b)) ADM_base

((c)) CWF ((d)) CWF_base

((e)) AIS ((f)) AIS_base

Figure 9.5: Distribution of average `waitTime`.

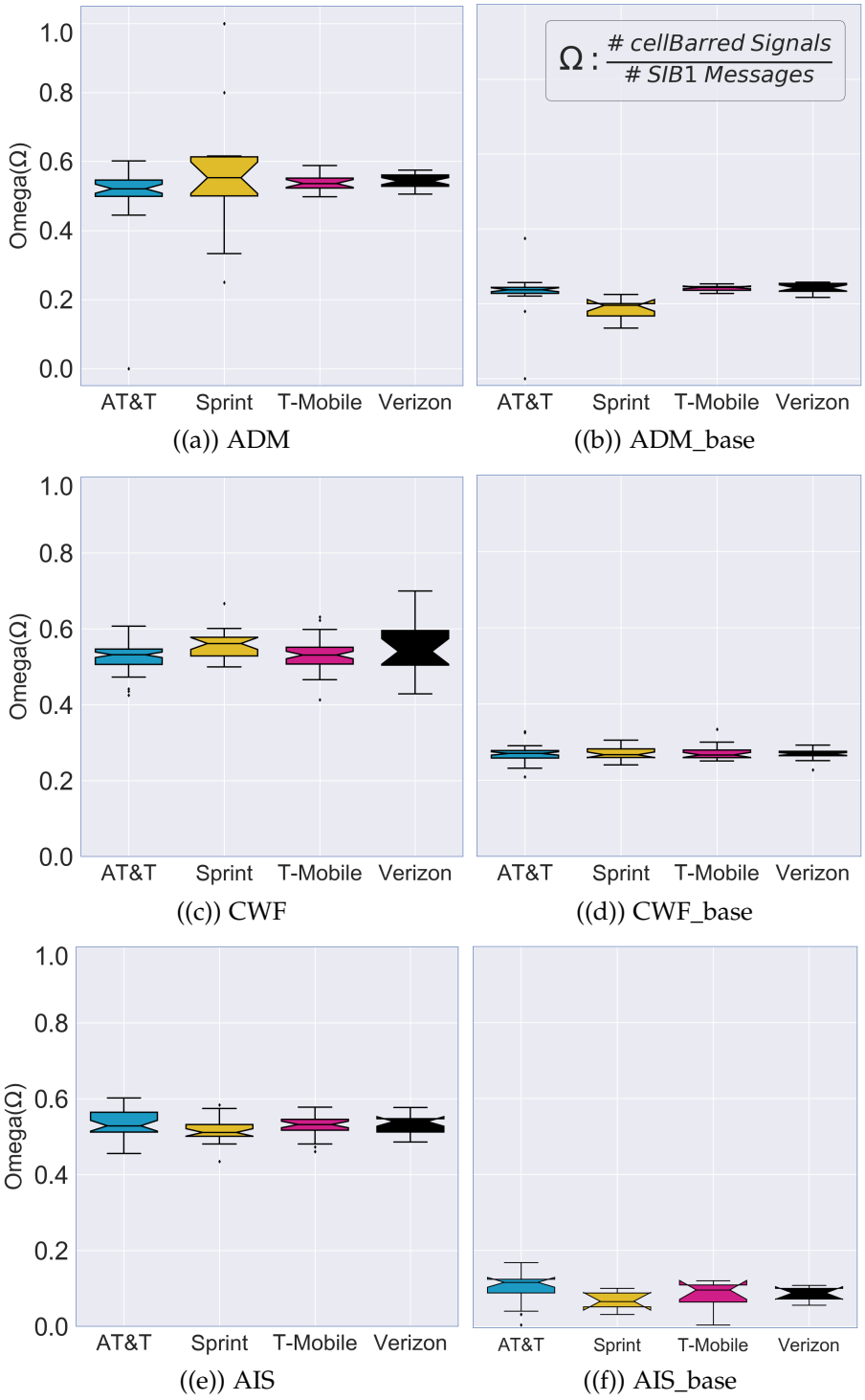
rather than broadcast unavailability (through cell barring messages), regardless of their proprietary overload mitigation schemes.

9.4.2 Congestion Detection through Quality of Service

Given the current density of LTE deployments, the massive amount of multimedia traffic on these networks calls into question the Quality of Service (QoS) of these flows. The extent of QoS optimization required from a network management context depends on the type of application being used. For instance, delay-sensitive Internet traffic, such as live streaming video, voice over IP, and multimedia teleconferencing, requires low end-to-end delay to maintain its interactive and live nature. On the other hand, on-demand gaming traffic depends on end-to-end delay and achieved throughput. One of the primary barriers to achieving usable QoS in LTE networks is high network utilization, which can cause congestion. In this portion of our study, we collect three QoS metrics (throughput, latency, and packet loss). For throughput tests, we download a 10 MB file from an AWS instance; packet loss is computed from the gathered packet traces. We use Hping3 to collect average RTT from the same AWS instance for latency. In this section, we evaluate those QoS metrics to study the effect of congestion resulting from overloaded LTE networks. Our analysis reveals stark differences in the performance of congested and baseline measurements. Results show: (i) $10\times$ – $21\times$ lower throughput, (ii) $2\times$ – $12\times$ higher latencies and (iii) $8\times$ – $11\times$ higher packet losses in congested locations.

Throughput

Throughput on mobile broadband is a crucial parameter that reflects the network's health. Comparing throughput across an overloaded network to its baseline



((a)) ADM ((b)) ADM_base

((c)) CWF ((d)) CWF_base

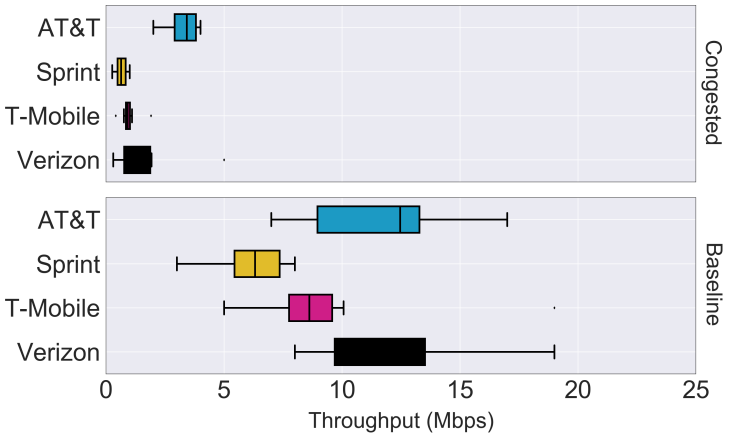
((e)) AIS ((f)) AIS_base

Figure 9.6: Omega (Ω) measure in thirty-second bins.

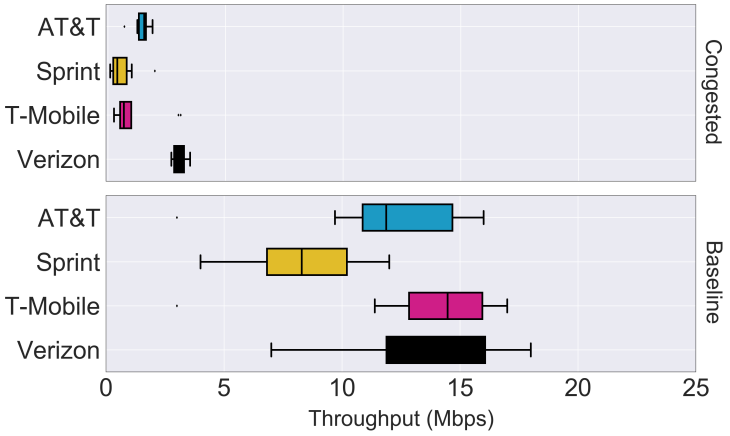
can reflect the extent of network congestion. Figure 9.7 shows the comparison of throughput measurements for the overloaded datasets (figures at the top) and their corresponding baselines (bottom figures). We observe that throughput decreases substantially during heavy overload conditions, suggesting network congestion. Across all locations, AT&T and Verizon fare better than their competitors under normal operating conditions, consistent with results from independent studies across the industry [210, 211]. Unsurprisingly, Sprint has the lowest throughput average across all our datasets, congested or otherwise. Network congestion leads to 24× reduction in throughput measurements, as shown in Figure 9.7(c). In the congested ADM dataset, we observe higher variability associated with the Verizon and AT&T networks. This suggests that congestion mitigation schemes employed by these networks are marginally more effective, as demonstrated by higher median throughput values in Figure 9.7. AT&T and Verizon maintain steady rates across all baseline datasets despite serving a disproportionate fraction of users (more than 55% contribution in LTE frame captures).

Latency

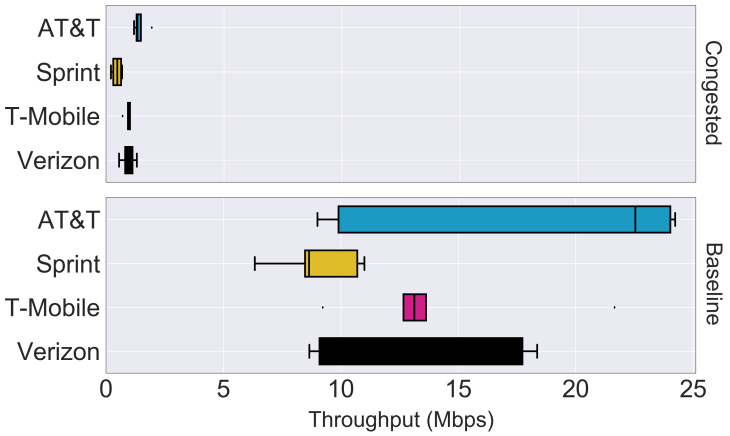
With the advent of LTE and 5G networks, stringent requirements have been imposed on latency and reliability [212] with claims by some operators to introduce ultra-low latency on "advanced" LTE networks [213]. Thus, consistent low round-trip time latencies indicate a well-functioning network [97]. We collect over 200 latency datapoints for each operator at every location (i.e., ADM, CWF, and AIS). Table 9.2 shows the average round-trip times across congested and baseline measurements. We learn that during overload conditions at ADM, average RTT almost doubles, which is reflective of network congestion. We observe elevated levels of congestion in CWF, which reports latencies as high as 14× (T-Mobile) in its baseline



a. ADM



b. CWF



c. AIS

Figure 9.7: Throughput measurements across all locations. Figures at the top are from congested datasets while figures at the bottom correspond to baseline measurements.

Table 9.2: Round-trip Times

Locations	ATT	Sprint	T-Mobile	Verizon
ADM	123 ms	112 ms	151 ms	138 ms
CWF	703 ms	3237 ms	857 ms	202 ms
AIS	191 ms	198 ms	134 ms	174 ms
ADM Base	62 ms	64 ms	62 ms	63 ms
CWF Base	63 ms	63 ms	61 ms	62 ms
AIS Base	59 ms	67 ms	63 ms	61 ms

measurements. Sprint's network appears exceedingly congested, with an average RTT 48× higher than the baseline. Another notable observation is AIS, where even with the densely populated users and higher demand that lead to the lowest average throughput (§9.4.2), the average RTT for AT&T, Sprint, and Verizon were about 3× the baseline values. This congestion level explains the meager average throughput achieved in Figure 9.7(c). These tests provide valuable insights into understanding the effect of overload in LTE networks, manifesting in congestion and subsequent degradation in user experience.

Packet Loss

Packet loss in cellular networks is more prominent than in wired networks [141]. Loss can happen due to network congestion and transmission errors [141]. We compare packet loss rate during crowded events with network overload (§9.4.1) and contrast these loss rates with those observed during instances of low network utilization. From Table 9.3 we infer that packet loss increases during overload conditions. We argue that this elevation in loss rate can be attributed primarily to congestion since our physical placement for data collection remains identical in ADM/ADM base and CWF/CWF base; in AIS base, we position ourselves ∼30 meters from our original

Table 9.3: Packet Loss Rate

Locations	ATT	Sprint	T-Mobile	Verizon
ADM	0.61%	1.44%	1.92%	1.63%
CWF	1.92%	3.47%	2.69%	0.86%
AIS	2.14%	1.73%	1.38%	1.15%
ADM Base	0.15%	0.35%	0.17%	0.22%
CWF Base	0.26%	0.14%	0.39%	0.10%
AIS Base	0.21%	0.16%	0.20%	0.08%

location. In addition, we compare the reference signal received power (RSRP) values logged during the tests within each pairing of our datasets (congested and baseline) to ensure uniformity in radio measurements. We do not claim to have eliminated all aspects of the wireless medium that could contribute to the loss rate. For instance, there could be temporary link failures or high bit-error rates. Instead, our focus is to eliminate apparent wireless channel discrepancies that could affect our measurements, such as differential RSRP values. We observe that Sprint's network experiences 25× more packet loss compared to its baseline in CWF, whereas other networks have escalations between 6× and 8×. In AIS, we see a mean increase of 10× over the baseline, whereas ADM reports a 6.5× escalation in packet loss.

9.4.3 Congestion Detection through Quality of Experience

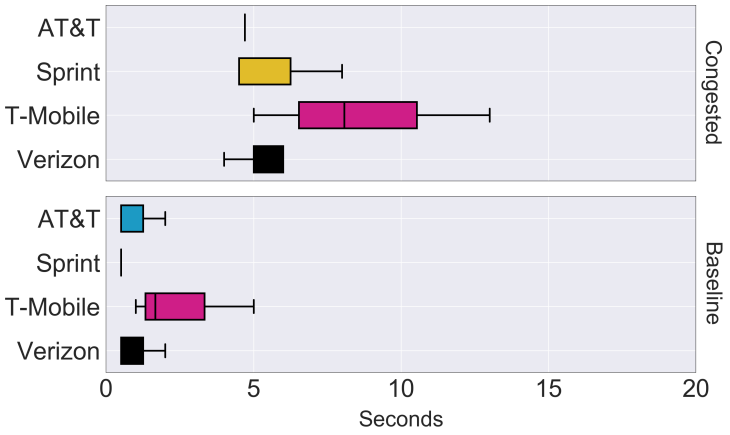
Quality of Experience (QoE) is one of the leading concepts for network management and performance evaluation in operational networks. Among the most relevant QoE-centric services consumed by end customers in mobile networks, Web surfing and mobile video take the prime spots [214]. In particular, video now represents over three-quarters of the global IP traffic [214]. In this section, we study the performance degradation introduced by congestion as overload increases on LTE networks. Our

results indicate: (i) 6× – 38× higher start-up delay, (ii) 3× lower video quality, (iii) 3× – 6× higher stall ratio and (iv) 33% – 56% lower residual buffer levels in the congested dataset.

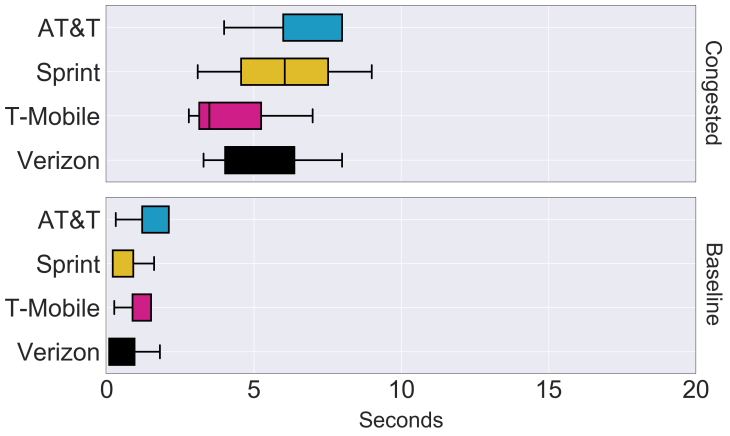
Video Streaming: YouTube

Start-up delay: Start-up delay is the time lag between user action to play video and video starting to play on the screen. This delay usually corresponds to how quickly the HTTP Adaptive Streaming (HAS) client can fill the threshold buffer required for playback. For instance, we observe diminishing throughput during congestion in Figure 9.7. Such scenarios would likely require additional time to download the same number of video chunks that go into the video buffer than in an uncongested network, leading to higher start-up delays. Here we note that the start-up delay does not convey any information about the video resolution chosen for playback. Figure 9.8 reports the delay incurred during our measurement campaign. Upon examination, we observe a significant increase in delay as overload increases (figures at the top), signaling heavy congestion in the networks. We see that T-Mobile and Sprint have the most heavy-tailed distributions (outliers on either end of the box), which indicate variable delay. This could be due to either variability in network throughput or client fallback to lower resolution video, possibly after failed attempts to achieve higher bit-rates (or fetch higher resolution chunks).

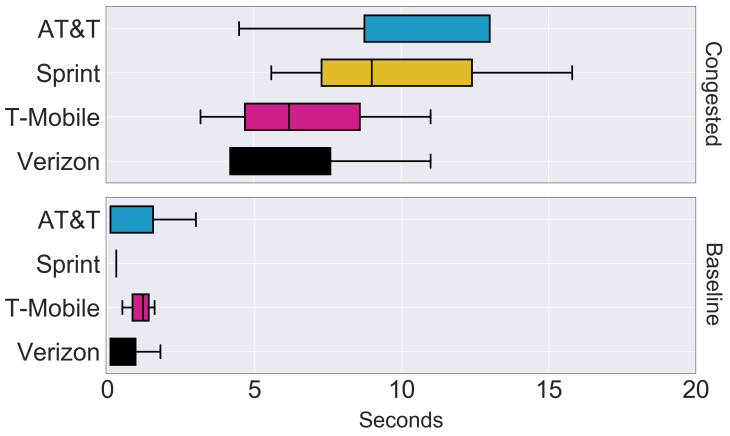
Video Quality: With the proliferation of high resolution displays on smartphones and tablets in the past few years, it is now possible for users to take advantage of high-definition videos on their devices, with some mobile devices that offer 4K ultra high-definition capability. Prior studies have illustrated that a drop in video resolution has a notable adverse effect on user experience, such as sustained frustra-



a. ADM

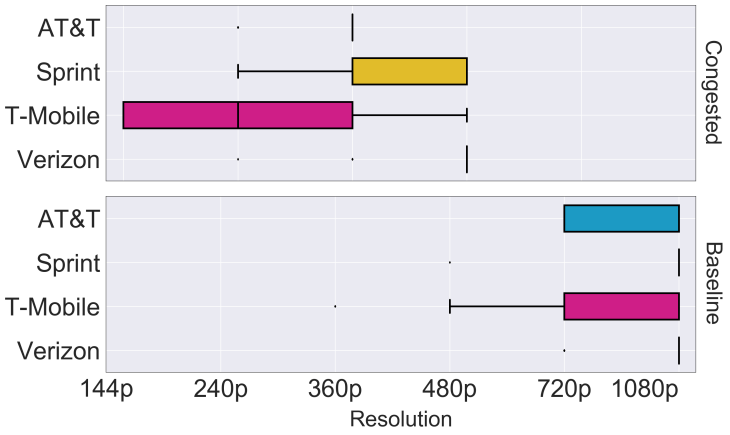


b. CWF

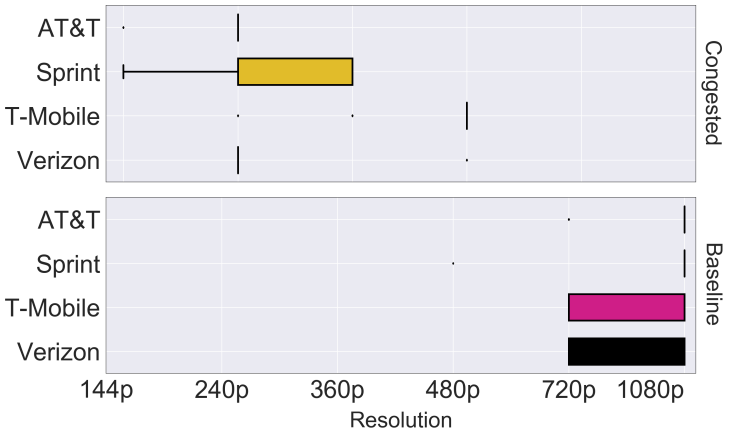


c. AIS

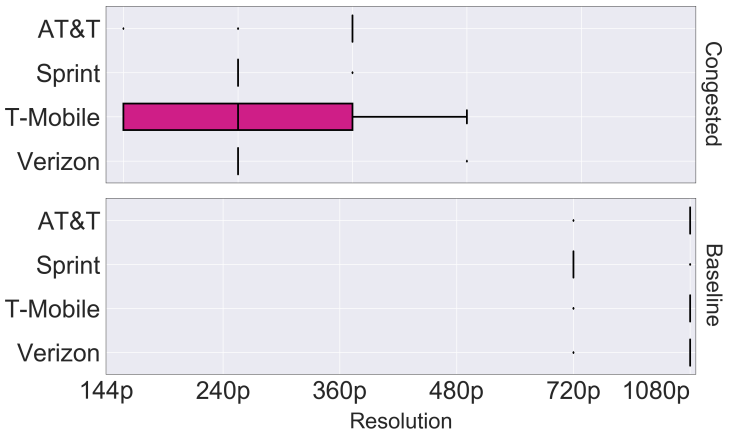
Figure 9.8: Start-up delay during YouTube streaming. Lower is better.



a. ADM



b. CWF



c. AIS

Figure 9.9: Achieved video quality during YouTube streaming. Higher is better.

tion [215]. Figure 9.9 depicts the playback resolution of the YouTube video, sampled at one-second granularity. During our measurements, we ensure that the video used for playback is uniform across all our datasets. However, the resolution is set to auto. Stated otherwise, final playback resolution and switches are dependent on network conditions and changes in congestion levels. While most of the baseline measurements (Figure 9.9, plots at the bottom) indicate near full-HD (1080p) rendering of the video, the congested dataset (plots at the top) reports a severe drop in resolution. AT&T and Verizon have near consistent resolution rendering during congestion, while T-Mobile and Sprint networks have wide variability. This is a significant finding. Variability in video resolution implies constant quality switches, which is usually perceived as an acute case of performance degradation in QoE [112, 215]. Overall, T-Mobile has the most significant number of quality switches across all locations (congested and baseline).

Stall Ratio: Re-buffering events on video streaming applications usually translate to unusable service [215]. Among other network artifacts, congestion can increase re-buffering events while streaming online videos [103]. If re-buffering happens, the user notices interrupted video playback, commonly referred to as *stalling*. The stall ratio is the amount of time the video stalls during the playback expressed as a fraction of total playback time, shown in Table 9.4. Although not all telecoms report stalling across the three locations (i.e., ADM, CWF, and AIS), those have a significantly higher ratio than their corresponding baselines. For instance, we see a 60$\times$ increase stall ratio on the Verizon network in CWF. Similarly, our analysis reveals a 30$\times$ and 3$\times$ increase on the T-Mobile network at AIS and CWF, respectively. AT&T and Verizon report a non-zero stall ratio across all congested datasets. Despite its poor performance in start-up delay and video quality, Sprint has the least stalled

Table 9.4: YouTube stall ratio percentage.

Locations	ATT	Sprint	T-Mobile	Verizon
ADM	4.56%	0%	0%	0.85%
CWF	2.3%	0%	3.07%	18.53%
AIS	3.41%	0.29%	7.27%	2.49%
ADM Base	0%	0%	0%	0%
CWF Base	0.68%	0%	0.98%	0.3%
AIS Base	0%	0%	0.22%	0.44%

video, with no stalls reported in either ADM or CWF locations.

Buffer Size: The streaming client employs a playout buffer or client buffer, whose maximum value is buffer capacity, to temporarily store chunks to absorb network variation. To ensure smooth playback and adequate buffer level, the client requests a video clip chunk by chunk using HTTP GET requests and dynamically determines the resolution of the next chunk based on network conditions and buffer status. When the buffer level is below a low threshold, the client requests chunks as fast as the network can deliver to increase the buffer level. The playback stalls when the buffer is empty before the end of the playback is reached. From the perspective of YouTube video playback, a session can contain two exclusive regions: buffering and playing. The buffering region is defined as the period when the client is receiving data in its buffer, but video playback has not started or is stopped. The playing region is the period when video playback is advancing regardless of buffer status. In the playing region, the video state can be either buffer increase, decay, or steady. Figure 9.10 shows the distribution of buffer size captured during YouTube streaming sessions. Congested locations demonstrate smaller buffer sizes than the baseline measurements. The median difference in ADM, CWF, and AIS is 17.73 seconds, 23.51 seconds, and 21.1 seconds, respectively. Verizon's median difference is

the lowest among all the operators we evaluate, at about 12.7 seconds. On the other hand, Sprint registers the widest variance with a median difference of more than 30 seconds.

Page Load Time

Web performance has long been crucial to the Internet ecosystem since a significant fraction of Internet content is consumed as Web pages. A considerable share of applications such as Web, e-mail, or just non-native social media access implies waiting times for their users, which is reflective of the responsiveness from the requested server. Responsiveness is also a function of network conditions, such as congestion [216]. Thus, end-user quality perception in interactive data services is dominated by Web page loading times; the longer they wait, the lower the user satisfaction [216]. Moreover, studies have shown that *perceived* time for users accessing the Web can be exceedingly magnified for actual chronological time, thus degrading the *perceived* performance even further [115].

Page load times are depicted in Figure 9.11. From our evaluation, we learn that overloaded eNodeBs experience higher congestion levels that lead to a stark contrast between ADM, CWF, and AIS load times and their respective baselines. T-Mobile stood out as the worst-performing network across all congested datasets. AT&T, Sprint, and Verizon exhibit deteriorating performance in CWF, which can be explained by excessive round-trip times detected in Table 9.2. In the experimental setup, we set the timeout value to 30 seconds. Our choice for this timeout value is derived from Shaik et al. [217], who empirically found that users tend to get tired of wait times by terminating their Web sessions typically after 10-20 seconds. We present the results of Web page timeouts in Table 9.5. Not surprisingly, T-Mobile produces the highest number of timeouts in all congested datasets. Our examina-

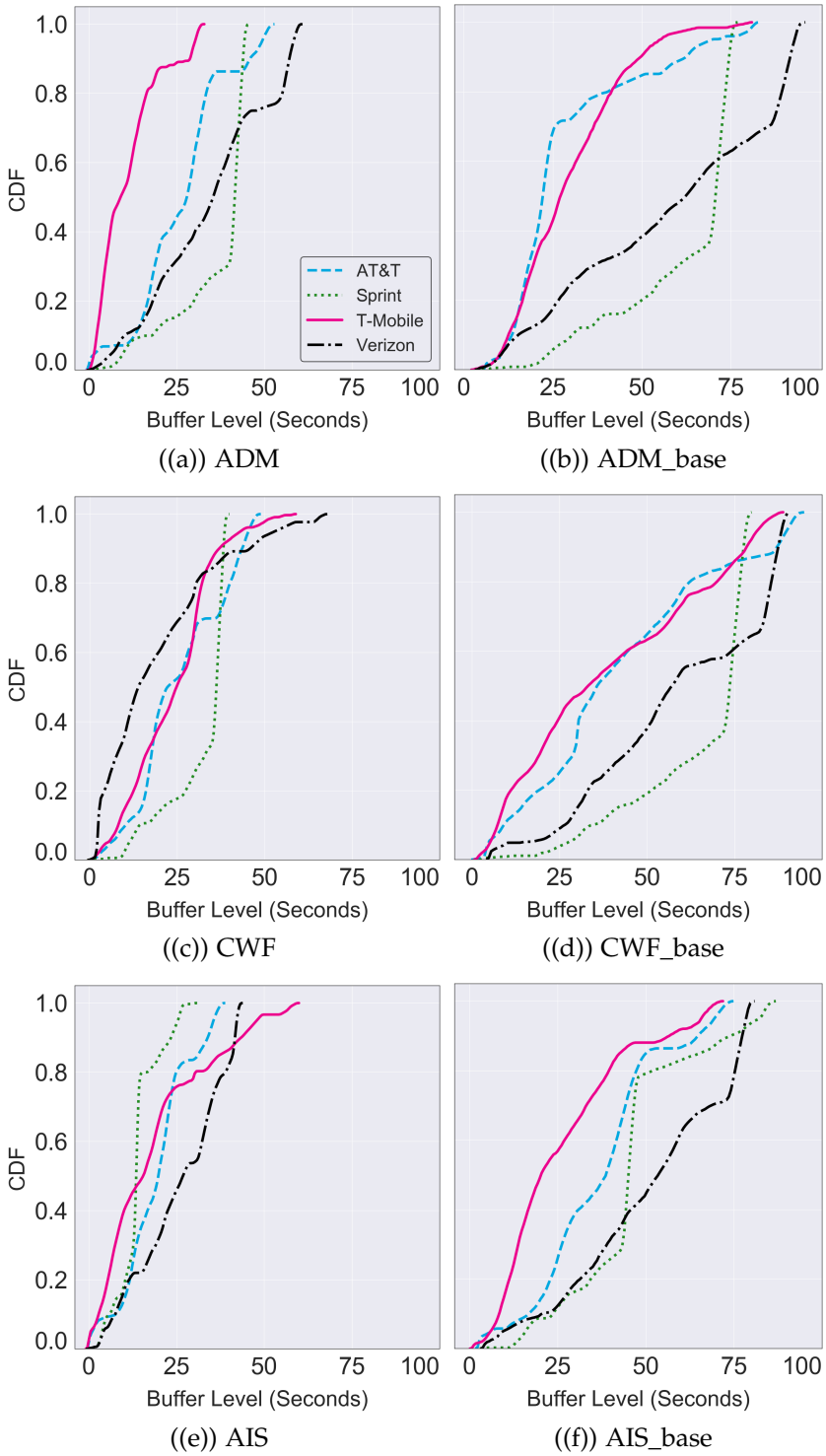
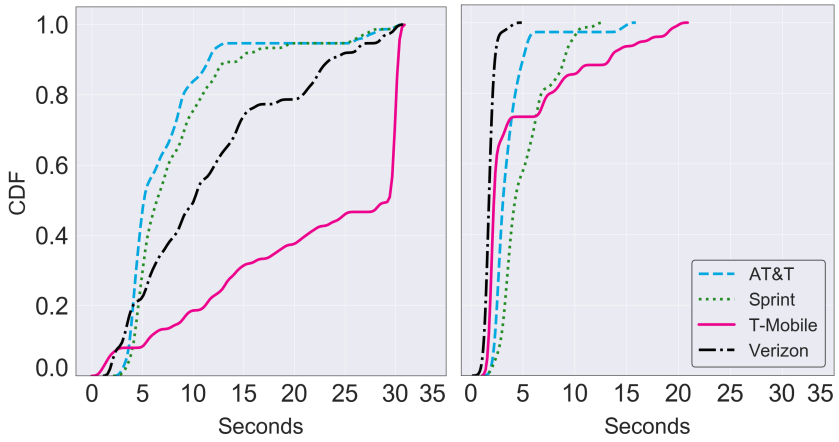
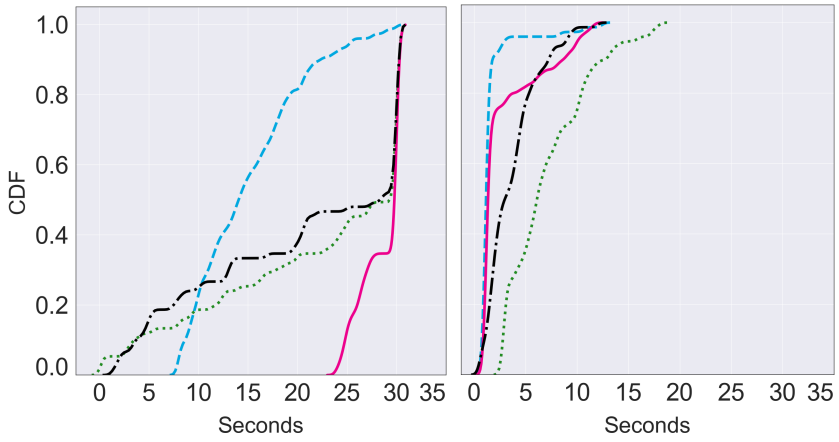


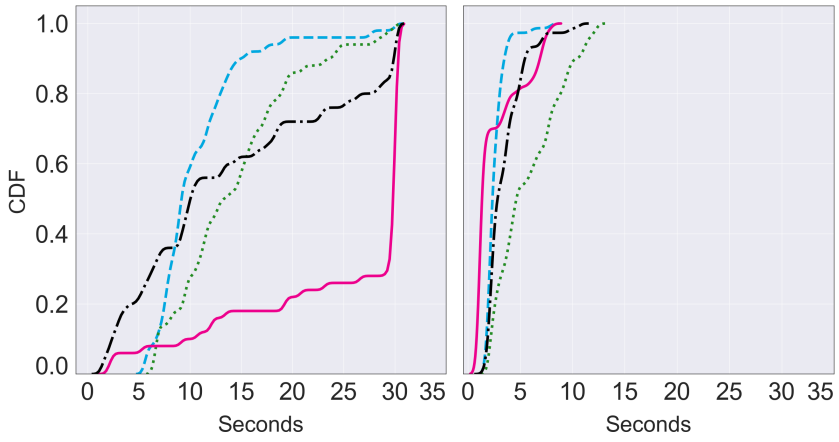
Figure 9.10: Cumulative distribution of buffer size (seconds) during YouTube test.



((a)) ADM ((b)) ADM_base



((c)) CWF ((d)) CWF_base



((e)) AIS ((f)) AIS_base

Figure 9.11: Page load times of Tranco top 25 websites on all operators.

Table 9.5: Page load time-outs

Locations	ATT	Sprint	T-Mobile	Verizon
ADM	5.33%	6.67%	50.67%	8%
CWF	9.33%	51.89%	64%	42%
AIS	6.67%	5.33%	73.33%	10.67%
ADM Base	0%	0%	0%	0%
CWF Base	0%	0%	0%	0%
AIS Base	0%	0%	0%	0%

tion reveals that the CWF dataset performs poorly than other congested datasets. In AIS, about three-quarters of websites on T-Mobile were either unreachable or could not load required objects before the timeout, indicating severe congestion across the network.

9.4.4 Discussion

We acknowledge that adding ground truth measurements from carriers would have provided another layer of validation. However, we could not obtain this data due to providers' strict policies against sharing client-related data. Instead, we use our validation methodology (using active measurements to characterize performance and detect congestion) to provide validation from the users' perspective. We believe that our active measurements produce results similar to those demonstrated by carrier ground truth data. Further, Lumos requires prior baseline measurements to infer the network condition (congested/overloaded). This is despite the 15–20% reject rate observed across our datasets. Further investigation is needed to ascertain whether these baseline measurements apply to other locations and networks.

9.5 Conclusion

In this work, we propose a novel method to assess congestion in nearby LTE eNodeBs, utilizing off-the-shelf hardware without requiring cooperation from the cellular provider. Our analysis offers convincing evidence that messages broadcast by the eNodeB can be used to detect network congestion by estimating cellular overload. In future work, we will explore how passive overload inference can be leveraged in a system for automated overload mapping using ground-based data collection and Unmanned Aircraft Systems (UASs), independent of collaboration from a cellular provider. Software-defined radios on UASs have been shown as effective tools for rapidly deployable LTE coverage mapping [35], and we are exploring expanding aerial capabilities to include overload estimation. Such tools can be leveraged by regulators and policymakers and allow targeted deployment of alternative communication channels. While Lumos accurately estimates network overload and congestion, we investigate various methods to infer the service quality at UEs through QoE-estimation in the next chapter.

Chapter 10

Too Late for Playback: Estimation of Video Stream Quality in Rural and Urban Contexts

10.1 Introduction

More than 60 million people reside in rural regions in the United States [218]. However, cellular deployment is often guided by economic demand, concentrating deployment in urban areas and leaving economically marginalized and sparsely populated regions under-served [24]. Few prior studies have focused on assessing mobile broadband in rural areas of the U.S.; there is a lack of accessible datasets that are comprehensive (including network-level and application-level traces) and representative and inclusive of rural demographics. As a result of the COVID-19 pandemic, the assessment of the quality of experience (QoE) for applications delivered over mobile broadband has become urgent as stay-at-home orders, and rapid movement to online schooling and work-from-home protocols increase the demand for applications that

are known to be sensitive to network quality, such as video streaming and interactive video chat [219]. As a result, communities without access to usable, high-speed broadband, such as many rural communities, are particularly disadvantaged [25, 26].

Unfortunately, the evaluation of user quality of experience for video streaming applications accessed over LTE in regions where people are most likely to be smartphone dependent [45, 24, 6] poses a significant scalability challenge. QoE metrics collection over LTE networks in a geographic area requires time and resource-intensive measurements for each network provider. As a result, experiments at a single geographic point can be lengthy. Moreover, in rural areas, obtaining LTE Internet measurements in places where people are likely to use mobile broadband (e.g., at their homes or along local transportation corridors) can be challenging [35], as places of interest are far apart (requiring more resource-intensive targeted measurement campaigns) and less densely populated (prohibiting representative crowd-sourcing measurement efforts). In this context, we ask the following research question: *How can we infer the QoE for video streaming applications over LTE at scale?*

While few existing datasets measure QoE in rural communities, many public and proprietary datasets report quality of service (QoS) metrics, such as reference signal received power (RSRP) or throughput. These metrics are typically reported independently and are measured over LTE networks in a wide range of locations throughout the U.S. and globally [33, 210, 34, 129, 220, 122]. We argue that the wealth of LTE-QoS data points across the U.S. represents a key resource that can be leveraged to assess QoE broadly. While measuring QoE at scale in LTE networks presents significant challenges, measuring QoS at scale in LTE networks has already been demonstrated to be feasible. Hence, *our goal and key contribution is a methodology that can leverage low-cost QoS measurements to predict QoE.*

A diverse set of network measurements representative of a wide range of condi-

tions is needed to study the correlation between mobile QoS and QoE performance. As such, we undertook an extensive measurement campaign to collect 16 datasets comprised of network traces from the Southwestern U.S. for four major telecom operators: AT&T, Sprint, T-Mobile, and Verizon. Our datasets vary along two primary axes: population density and network load. We collected LTE network measurements within multiple rural and urban communities to obtain data from varied population densities. For variable network load, we collected LTE network traces from crowded events in urban locations that resulted in atypically high volumes of network utilization [90] and, consequently, congestion. As a baseline, we also collected traces from the exact urban locations during typical operating conditions. Our datasets have broad spatial and temporal variability but can be classified into three primary categories: under-provisioned (rural), congested (congested urban), and well-provisioned (baseline urban).[1] We leverage these varied datasets to demonstrate the generality of the inference method. Our analysis shows that predictive models can be used to infer video QoE metrics using low-cost QoS measurements so that QoE can be more easily and scalably determined within difficult-to-assess regions.

Our key contributions and findings include:

- We collected sixteen measurement datasets[2] from twelve locations through an extensive ground measurement campaign within the Southwestern U.S. Our data points represents three different network conditions: under-provisioned (rural), congested, and well-provisioned urban, and include over 32 Million LTE packets. (§11.2);

- We develop and evaluate a comprehensive set of predictive models that infer

[1]Through extensive analysis, we verified that our datasets are representative of the network characteristics we anticipated: well-provisioned, congested, and under-provisioned. A detailed analysis is provided in Chapter 4.

[2]The subset of our dataset that we have permission to release is available at [221].

video QoE from low-cost QoS measurements such as RSRP and throughput. Our analysis reveals that predictive models can infer video QoE with an accuracy of at least 80% across all locations and network types (§10.3);

- We validate our models across multiple video types from various genres. Further, we demonstrate the utility of low-cost RSRP measurements for inferring video QoE (§10.3).

10.2 Methodology and Datasets Overview

QoS metrics, such as received signal strength, latency, throughput, and packet loss, capture the state of network connectivity. However, while QoS indicates network state, there can be a disconnect between QoS and user experience. QoS network metrics are not Pareto-optimal; one element can get better or worse without affecting the other. Consequently, estimation of user experience requires incorporating multiple network measures, which may be unique to time, space and application. Note that while the definition of QoE can vary depending on the vantage point from which measurements are taken, we only focus on application-level QoE. Our measurements are active end-user device/passive user as defined in [222].

10.2.1 QoS and QoE Metrics

This section describes the QoS and QoE metrics we collected (and estimated) for this measurement study, as summarized in Table 10.1.

Quality of Service Metrics: We collect *reference signal received power (RSRP)* and *throughput* synchronously on the same user equipment (UE). RSRP is defined as the linear average over the power contributions (in Watts) of the resource elements

Table 10.1: Overview of QoS and QoE metrics at each location, aggregated across available providers.

Type	Metric	Test Interval	Number of Datapoints	Tools
QoS	RSRP	1 second	2160	Network Monitor
	Throughput	1 second	2160	iPerf
QoE	Video resolution	1 second	2160	Selenium, iframe API
	Resolution switches	1 second	2160	Selenium, iframe API
	Rebuffering events	1 second	2160	Selenium, iframe API

that carry cell-specific reference signals within the measurement frequency bandwidth [92] and, as illustrated by [95], is widely accessible through mobile operating systems. We record instantaneous RSRP readings from the UEs every second through the Network Monitor application [79]. We measure throughput by fetching a pre-specified 500 MB file from an AWS instance in Virginia using iPerf over TCP to download the file. The large file size allows the data traffic to fill the pipe and minimize the effect of a slow start. We log the packet traces at the client during the iPerf tests to sample throughput at 1-second intervals.

Quality of Experience Metrics: We focus on streaming video, currently the most heavily used QoE-centric service in mobile networks [98]. Internet video streaming services typically use Dynamic Adaptive Streaming over HTTP (DASH) [99] to deliver a video stream. DASH divides each video into time intervals known as segments or chunks, encoded at multiple bit rates and resolutions. We gather two QoE metrics to analyze video stream quality: *resolution switches* and *rebuffering events*. For resolution switches, we compute the number of consecutive samples with a different resolution as a percentage of the total number of samples collected during the video. We measure at one-second granularity, capturing resolution switches between video *chunks* that are typically 4–5 seconds long [102]. Finally, rebuffering occurs when the video pauses while the application buffer waits to accumulate enough content to

resume playback. We record the video state (rebuffering event or normal playback) every second.

10.2.2 Measurement Suite

We run our measurement suite on Lenovo ThinkPad W550s laptops, each tethered to their own Motorola G7 Power (Android 9) via USB to measure cellular performance. The cellular plans on all our cellular user equipment (UE) have unlimited data and are hot-spot enabled to effectively achieve the same level of performance as we would on the mobile device. We run our measurement suite on laptops tethered to phones; this configuration gives us the same application performance while facilitating ease of programming, data extraction, and unification of application-level measurements.

We choose YouTube as the streaming platform because of its popularity in the U.S., capturing over 88% of the mobile market [107]. To collect video QoE metrics, we run a 3-minute clip of a Looney Tunes video [223] three times across each of the four LTE providers at each location; we exclude from our results the sessions that experienced playback errors during execution. We chose this particular video due to its mix of high and low action scenes, which result in variable bitrates throughout the video (typically, high action scenes have a higher bitrate than low action scenes). After testing multiple playback duration, we observed that a 3-minute window was adequate for the playback to reach a steady state while long enough to capture rebuffering and resolution switches. To infer video QoE, we collect the input features (RSRP and throughput) synchronously on a separate device so as not to bias the video streaming measurements. Synchronous measurements of throughput, RSRP, and QoE metrics are required to train learning algorithms to infer video QoE for a future time instance. We use different servers for throughput and YouTube tests to

obtain concurrent QoS and QoE measurements. Our setup reflects the real-world scenario where throughput test servers and YouTube servers are separate while simultaneously affected by varying conditions from *within* the cellular network [97]. In LTE, each bearer (connection from a UE) enjoys a relatively isolated data tunnel before the egress from the packet gateway, located inside the core [224]. This reduces contention among UEs competing for resources at a single eNodeB, and as a result, we can accurately record QoS and QoE metrics on two separate devices.

To execute this experiment, we first automate the loading and playback of the YouTube video on the Chrome browser using Selenium [105]. The video resolution is set to auto. Then we use YouTube's iframe API [108] to capture playback events reported by the video player. The API outputs a set of values that indicate player state (not started, paused, playing, completed, buffering) using the getPlayerState() function. The API also provides functions for accessing information about playtime and the remaining buffer size.

10.2.3 Description of Datasets

We collected 16 datasets from 12 locations across the Southwestern U.S. Eight of the datasets were collected from rural locations with sparse cellular deployment.

An additional eight datasets were collected from four urban locations. In each urban location, we collect two datasets: one during a large event or gathering, in which we expect cellular network congestion to occur (these datasets are marked with **_Cong**); and a second during typical operating conditions. We call the latter dataset the baseline for that location (these datasets are marked with **_Base**). Hence, our 16 traces are broadly classified into three categories: rural, congested urban, and baseline urban. The details of each dataset are summarized in Table 10.2. The designation of each location as rural or urban is based on Census Bureau data [38].

Table 10.2: Summary of Datasets

Location	Date	# LTE Packets	Type	Carriers*
Rural_1	May 28 2019	3.18 Million	Rural	V,A,T,S
Rural_2	May 29 2019	1.38 Million	Rural	V,T
Rural_3	May 28 2019	2.03 Million	Rural	V,A,T,S
Rural_4	May 30 2019	2.16 Million	Rural	V,A,T,S
Rural_5	May 30 2019	2.27 Million	Rural	V,A,T,S
Rural_6	May 31 2019	2.33 Million	Rural	V,A,T,S
Rural_7	May 31 2019	1.26 Million	Rural	V,T
Rural_8	Jun 01 2019	2.83 Million	Rural	V,A,T,S
Urban_1_Cong	Sep 22 2019	2.25 Million	Urban, Congested	V,A,T,S
Urban_1_Base	Sep 28 2019	1.92 Million	Urban, Baseline	V,A,T,S
Urban_2_Cong	Sep 29 2019	2.51 Million	Urban, Congested	V,A,T,S
Urban_2_Base	Sep 30 2019	1.97 Million	Urban, Baseline	V,A,T,S
Urban_3_Cong	Sep 21 2019	2.65 Million	Urban, Congested	V,A,T,S
Urban_3_Base	Sep 30 2019	2.13 Million	Urban, Baseline	V,A,T,S
Urban_4_Cong	Sep 25 2019	2.18 Million	Urban, Congested	V,A,T,S
Urban_4_Base	Sep 26 2019	2.08 Million	Urban, Baseline	V,A,T,S

*This column lists the mobile carriers present in each data set (some areas had no coverage for particular network operators). V: Verizon, A:AT&T, T:T-Mobile, S: Sprint.

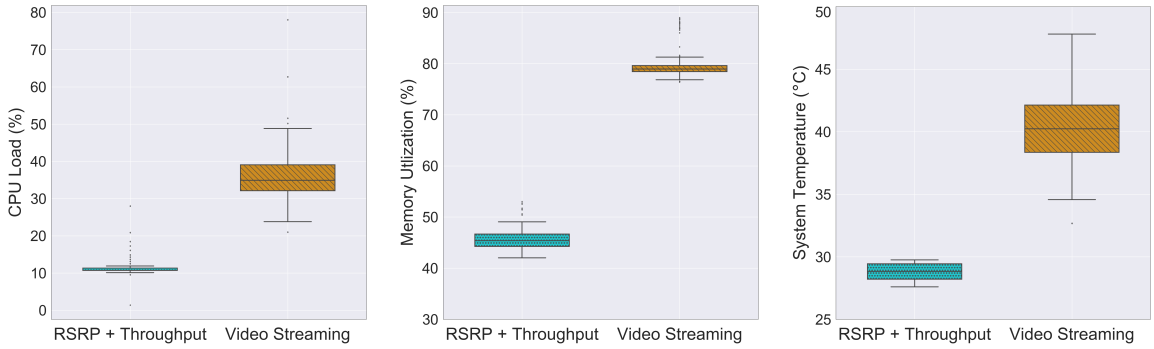
We collect and analyze over 32.7 Million LTE packets through these measurement campaigns. Note that the “Number of Datapoints" column shown in Table 10.1 indicates the QoS/QoE datapoints gathered by the application, while the “# LTE Packets" column in Table 10.2 refers to the number of packets collected in the trace files.

10.2.4 Video QoE Measurement Scalability Challenges

Collection of *ground-truth* cellular network measurements, as we explore further in §10.4, is a challenging task for multiple reasons. First, it requires the physical placement of the measurement device at the location to be studied. While many large, publicly accessible datasets incorporate some QoS measurements, QoE measurements, particularly in remote regions, are much more difficult. Second, gathering ground truth data to assess video QoE requires an active connection to stream a large encoded video file. This consumes a substantial amount of bandwidth, compu-

tational power, memory, and battery, due to the simultaneous use of LTE modems, display, CPU, and GPU [225] on the user device. For instance, streaming applications consume memory to load the video and require accelerated processing to decode and display the stream from the video server. Unlike QoS metrics, which can often be collected in the background through execution by backend scripts, the high resource cost of QoE measurements for the end-user makes this data difficult to crowd-source. In Figure 10.1 we show the resource consumption during one hour of RSRP and throughput (QoS) measurements, compared to one hour of video streaming (QoE), on our data collection phones. As seen in the figure, the resources consumed by the QoE measurements were significantly higher, preventing background data collection and draining the device battery more rapidly.

Rural regions span large geographic areas with terrain that is often hard to access. QoS data from public sources already struggle to cover these areas. In particular, crowd-sourced datasets are data-rich in regions with higher density populations. These regions tend to be either urban areas or other areas frequented by travelers (i.e., highways, national parks, etc.). Rural communities, by contrast, with their lower population densities, are often under-represented in crowd-sourced datasets. Yet it is precisely these regions where under-provisioned networks typically exist and hence



a. CPU Load b. Memory utilization c. System temperature

Figure 10.1: Device resource consumption during either RSRP and throughput measurements only, or during video streaming.

where data is urgently needed. We need a method to improve QoE measurement scalability to assess QoE in these remote areas effectively. We address this challenge in the next section, where we show how predictive models can use the less expensive QoS measurements to infer QoE for streaming video on mobile broadband networks in various environments.

10.3 Inferring QoE Metrics for Video

As discussed in §10.2.4, the collection of QoS measurements is less resource consumptive and hence more scalable than video QoE measurements. We now describe our approach to infer QoE metrics for video streaming sessions using low-cost QoS metrics.

10.3.1 Learning Problem

Our learning problem aims to infer QoE metrics using a sequence of throughput and RSRP (QoS metrics) data input. The objective is to build models with appreciable performance that would work in various network conditions and different region types (e.g., rural and urban locations). These models could be used to predict application QoE (in our case, video streaming) at a particular location. We use supervised learning to train two different binary classifiers. The first classifier infers whether the video's state is stalled or normal; the second infers whether there is any change in video resolution. Both models perform the classification task every second.

Input: The learning model takes a sequence of RSRP and throughput values as input. Both of these metrics are low-cost measurements and easily accessible. Given how adaptive bitrate (ABR) video streaming players operate, the changes in throughput and RSRP values have a delayed impact on QoE metrics. For example, a decreased

available throughput will force the video streaming player to use the buffered data before stalling.

As part of feature engineering, we had to determine how many RSRP and throughput values to use as input for the learning model. Intuitively, the use of longer sequences will improve accuracy. However, longer sequences also increase the learning model's complexity, which requires more training data to avoid over-fitting. After varying $n = 0 \rightarrow 180$ (total playback time of a session), we found that using a sequence of *three* throughput/RSRP values enabled us to strike a balance between model complexity and accuracy. A typical approach to assessing throughput would be to log continuous measurements for a long time and analyze the resulting mean/-mode of the distribution. However, our results (§10.3.3) indicate that we can infer the video quality from only a 3-second sample. This has the added benefit of reducing the resource utilization at the client device, such as data consumption and battery drainage, while accurately inferring the video stream quality.

Output: We train two separate binary classifiers to predict the video state and change in resolution at the granularity of one second. Predicting QoE metrics at such fine granularity enables opportunities to infer QoE with limited training data. Given the input features, our models infer how likely it is for the video stream to experience either a video stall or a resolution change in the next instant.

Training data: Our dataset consists of 32,596 data points. Each data point has input values: a sequence of three RSRP and throughput values, as well as two boolean labels: video state (playing or stalled) and resolution switches (yes–resolution will change; no–resolution will not change). We collected this dataset through our measurement campaign by conducting 181 video streaming sessions across multiple locations (§10.2.3). For each classifier, we label the output training samples into either of

Table 10.3: Breakdown of training and test set samples for both classifiers.

		Training Set		**Test Set**	
Classifier Type	Target Metric	Class 0	Class 1	Class 0	Class 1
Classifier 1	Rebuffering Event	22,175	642	9,504	275
Classifier 2	Resolution Switching	22,490	327	9,639	140

the two classes: class 0 is when playback is normal and devoid of any event (rebuffering or resolution switch), and class 1 is when there is an event. We performed the classification task by splitting the entire dataset into a ratio of 70:30 training to test sets, as described in Table 10.3. We split the overall training dataset into training and validation sets (80:20). We chose the samples proportionate to the size of each dataset category (rural, congested urban, and baseline urban). We present the models' performance per location, train the models on specific locations, and test on others not included in the training. We do not make any distinctions between operators since an operator-agnostic evaluation is a more comprehensive reflection of coverage and QoE at a particular location.

10.3.2 Learning Algorithm

We now present the learning models we used for the learning problem, our model training approach, and the method for addressing the inherent class imbalance.

Learning models: We trained a wide range of off-the-shelf classifiers for this learning problem to identify the classifier that strikes the best balance between performance (precision, recall, etc.) and generalizability. First, we trained simpler classifiers, such as gradient boosting [226], bagging [227], random forest [228], ARIMA [229], AdaBoost [230], etc. These classifiers offer better generalizability at the cost of performance. We also trained neural-network (NN)-based classifiers, such as a convolu-

tional neural network (CNN) [231] and recurrent neural network (RNN) [232] (in particular, LSTMs [233] and GRUs [234]), that offer higher accuracy but require considerable training data to avoid over-fitting.

Setup: We ran all the classifiers on a local machine that runs Ubuntu 18.04, powered by a 4-core i7-7700 CPU (3.60 GHz) with 64GB RAM and 8GB NVIDIA RTX 2080 GPU. We implemented the simpler classifiers using the scikit-learn 0.21 [235] library of Python, and NN-based models using Keras with Tensorflow backend [236]. We used four fully-connected layers for the NN-based classifiers. For *RNN-LSTM-Focal* (see Table 10.4), the network utilized 64, 32, and then 16 hidden neurons, in addition to a final output layer with hyperbolic tangent activation function. We used Grid Search [237] to determine the ideal hyper-parameter configuration for each neural network. To avoid over-fitting, we use a dropout of 0.4 while training with the Adam gradient descent optimizer [238]. We ran the RNN-LSTM model for 120 iterations with a batch size of 64.

Class-imbalance problem: As rebuffering and changes in the resolution are rare, most of our data points are normal, i.e., they do not have any rebuffering or resolution switching events. As a result, our dataset has the class-imbalance problem, typical for most anomaly detection problems. To address this issue, we applied the sampling technique SMOTE [166] to balance the classes artificially. However, such an approach reduces the number of data points we can use for training the classifier, affecting the accuracy. With SMOTE, we observed no improvements in accuracy with simpler learning models (e.g., SVM, random forest, etc.) and lower accuracy for NN-based classifiers. Therefore, for the NN-based classifiers, we adapted a new technique that has proven to increase classification accuracy in datasets that suffer from the class-imbalance issue for the object detection problem [167]. This technique addresses the

class-imbalance problem by reshaping the standard cross-entropy loss to lower the weights for the majority class [167]. It also introduces the concept of *focal loss* that prevents the majority class from overwhelming the classifier during training. The focal loss can be represented as:

$$FL(p_j) = \alpha(1 - p_j)^{\gamma} log(p_j) \tag{10.1}$$

Here, FL is the focal loss function, and p_j is the softmax probability of the j^{th} class for a particular observation. α and γ are two regularizing parameters. This loss function adds more importance when the network predicts a minority sample as opposed to the overly represented sample—making it ideal for performing classification on an imbalanced dataset.

10.3.3 Results

We now present the performance of the different classifiers we used for this learning problem. We also quantify their performance across different locations and video types for those that performed well. Finally, we quantify the contribution of an LTE-specific QoS metric, RSRP, in improving the accuracy of our learning models.

Performance: We analyze the performance of learning models in terms of accuracy, precision, recall, and training time. Table 10.4 summarizes the performance of all classifiers we explored. We observe that the accuracy of the rebuffering-event classifier is better than the resolution-switching one, as depicted in Figure 10.2. This difference is attributable to the smaller number of anomalous data points (resolution switches) in the data (see Table 10.3). In terms of accuracy, *RNN-LSTM-Focal* performs best. This is expected as this model makes the best use of the sequence

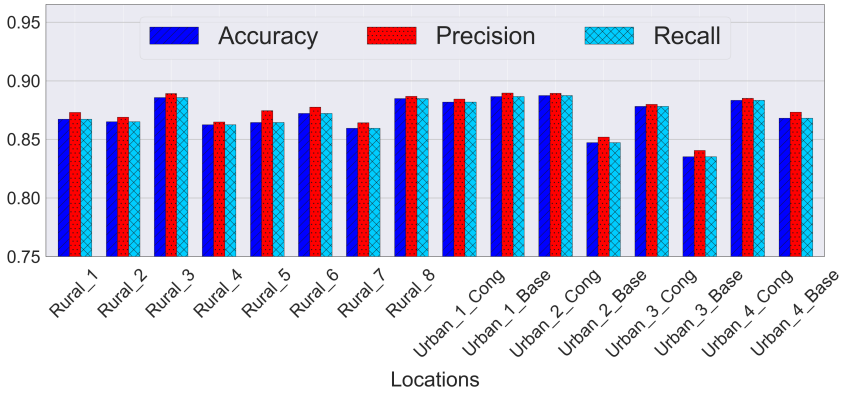
Table 10.4: Performance metrics of the classification models.

	Rebuffering Events			Resolution Switching		
Models	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Boosting	0.87	0.88	0.88	0.84	0.85	0.84
Bagging	0.80	0.82	0.82	0.71	0.73	0.72
Random Forest	0.85	0.87	0.86	0.79	0.80	0.80
ARIMA	0.81	0.81	0.81	0.77	0.78	0.78
Decision Trees	0.80	0.80	0.98	0.75	0.75	0.75
Extra Randomized Tree	0.77	0.78	0.77	0.72	0.73	0.72
AdaBoost	0.62	0.60	0.63	0.51	0.55	0.53
Support Vector Machine	0.72	0.72	0.73	0.70	0.71	0.70
K-nearest neighbors	0.60	0.56	0.62	0.58	0.57	0.49
CNN	0.72	0.73	0.73	0.68	0.69	0.69
CNN - Focal	0.84	0.85	0.84	0.81	0.81	0.81
RNN - LSTM	0.82	0.83	0.83	0.80	0.79	0.80
RNN - LSTM - Focal	0.89	0.89	0.89	0.86	0.86	0.87
RNN - GRU	0.82	0.82	0.84	0.80	0.82	0.82
RNN - GRU - Focal	0.86	0.86	0.85	0.83	0.84	0.84

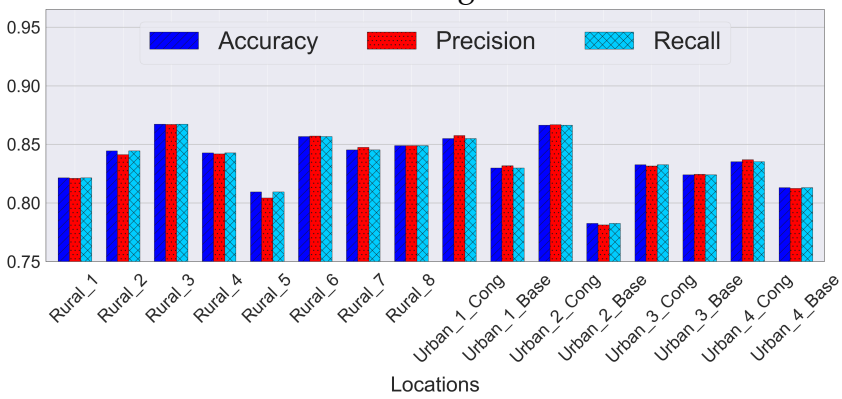
of throughput and RSRP values and is best suited to handle the class imbalance problem. On the other hand, though *RNN-LSTM-Focal* has the highest accuracy, the accuracy gains are marginal when compared to simpler learning models, especially *Boosting*. Given these marginal gains and the complexity of training NN-based classifiers (5 vs. 214 seconds), we use the *Boosting* classifier to characterize the performance across different network and video types.

Generalizability: We now quantify the generalizability of the *Boosting* classifier. First, we show how its performance varies across different network types. Figure 10.2 depicts the performance of inferring video rebuffering using *Boosting* at each location. We observe that the performance differences across different network types are marginal ($< 2\%$ deviation between categories). We saw similar trends for the *Boosting*-based classifier when inferring resolution switching.

Our initial measurements only collected the QoE metrics for the Looney Tunes video. We collected the QoS/QoE data for 108 additional video streaming sessions (48,825 new data points) at our research facility (baseline-urban) to verify that our re-



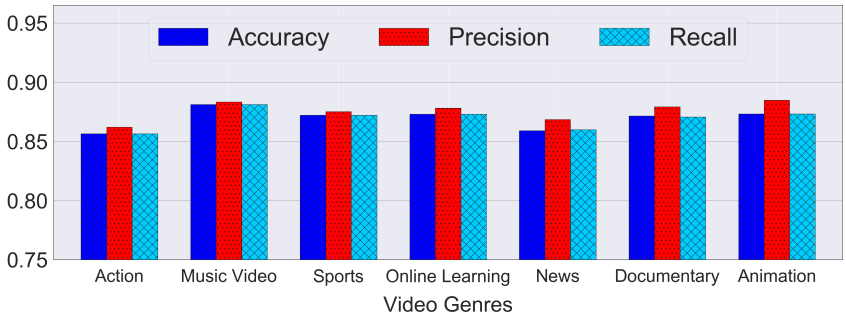
a. Rebuffering events



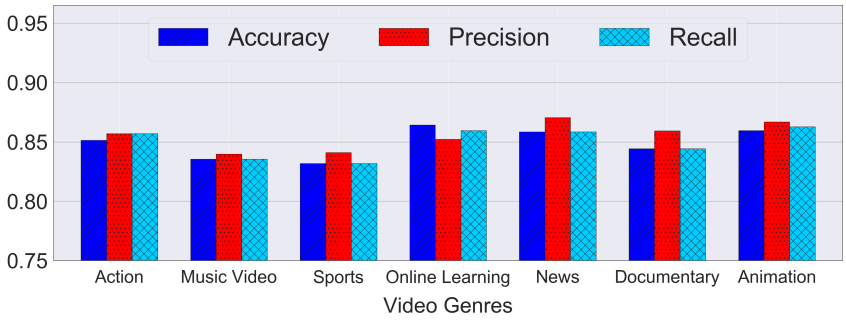
b. Resolution switching

Figure 10.2: Performance of *Boosting* across different locations.

sults generalize to other video types. We selected 18 different videos from seven genres: action (trailers/movie clips), music videos, sports, online learning content, news, documentary, and animation (including the original Looney Tunes video) [239]. We selected the top trending videos for each genre. Given that the videos were of varying duration, we capped each measurement to a maximum of ten minutes. We streamed each video over three different telecom providers (AT&T, T-Mobile, and Verizon); we could not obtain Sprint measurements because of the closure of Sprint retail outlets due to the COVID-19 pandemic. Figure 10.3 shows the performance of *Boosting* for both video rebuffering and resolution switching. We observe marginal variations ($< 1.5\%$ and $< 3\%$ deviation for rebuffering and resolution switching, respectively)



a. Rebuffering events



b. Resolution switching

Figure 10.3: Performance of *Boosting* across different video genres.

in accuracy across different video genres, implying that our learning model generalizes reasonably well to different video types. Note that we do not claim that these results generalize for other video players (e.g., Hulu, Netflix), client platforms, or devices; we plan to quantify the performance of our learning models for other platforms, devices, and non-YouTube videos in the future. Finally, we do not claim to have developed models that generalize across other locations or network conditions – rather, we use this study to demonstrate the feasibility of inferring video QoE *at scale* within a limited but diverse dataset.

Ablation Study: We performed an ablation study to understand better the impact of an LTE-specific metric (i.e., RSRP) in inferring QoE metrics. Figure 10.4 compares the accuracy of the *Boosting* classifier in inferring rebuffer events with and without the RSRP values. We observe that the average increase in accuracy, with RSRP as an

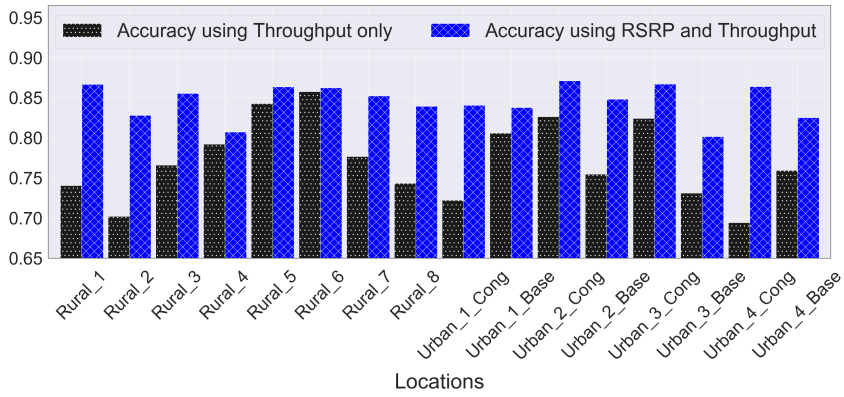


Figure 10.4: Inferring video rebuffering using *Boosting* with and without RSRP as an input feature.

input, is 9.28%, while the maximum gain is 18.61%. This result could be attributed to the exposition of the relationship, by the non-linear models, between RSRP and throughput to successfully identify the target metrics at any given location. This study highlights the importance of LTE-specific RSRP measurements in accurately predicting rebuffering and resolution switching.

10.4 Related Work

Prior work most similar to ours, which focuses on quantifying the user experience, typically infers the QoE of video streaming from QoS of fixed broadband networks [240, 241, 242]. In contrast, our work focuses on mobile broadband, which often exhibits a wide variation in performance over time and space. Some past work on mobile broadband, such as [243, 244, 245, 246], has examined metrics solely from the application and network layers. [247, 248, 208, 102, 112, 209] require direct access to (encrypted or unencrypted) network traffic to infer video QoE. In contrast, our approach is independent of network traces and incorporates low-cost signal and throughput measurements for rapid QoE prediction. Few publicly available QoS datasets in-

clude synchronous RSRP measurements. [249, 122, 250] analyze network traces that contain performance indicators captured during streaming sessions and experiment metadata from mobile broadband networks. All of these datasets, however, have limited types of datapoints (primarily from dense, urban locations); the datasets have minimal to no measurements from networks that are under-provisioned or located in remote regions. We believe it is challenging to utilize existing prior datasets (primarily urban scenarios) to evaluate diverse network conditions in the context of the measurements examined in this work, either due to the non-overlapping and non-scalable nature of prior measurements or the lack of a comprehensive and representative dataset. Further, the accuracy of our models, given the inexpensive measurements, indicates the feasibility and scalability of our approach.

Prior work focused on charting the relationship between RSRP and QoE has important limitations. For instance, [94] presents a mapping of RSRP and video QoE derived using only simulated experiments. The authors of [251] explore the effect of radio link quality, such as RSRP, on streaming video QoE. The presented results are limited in scope as their setup streams a custom video hosted on their server; by omitting evaluation of a popular streaming service, such as YouTube or Netflix, the work does not accurately capture the application and network performance experienced by actual users. [252] undertakes a study similar to ours, with a modest dataset that is limited to a small portion of a local transit route and thus difficult to generalize.

10.5 Conclusion

We collect 16 datasets with widely varying performance profiles through an extensive measurement campaign. Our dataset includes representation of: i) the variability

of mobile broadband performance as a consequence of either sparse deployments or network congestion, and ii) the communities most likely to depend on mobile broadband (rural areas). Our analysis highlights the challenges of quantifying QoE metrics at scale, particularly in remote locations. To address this challenge, we develop learning models that use low-cost and easily accessible QoS data (LTE-specific RSRP and throughput) to predict QoE metrics. Our models can be generalized to video content from different genres, as well as to other locations that share network characteristics similar to those of our dataset. The observed efficacy of the models indicates that video QoE can be more easily and scalably determined within difficult-to-assess regions using low-cost QoS measurements. For instance, given the increased load on video streaming platforms during COVID-19 [219], cellular operators could employ our approach to detect sectors with possible bottlenecks without relying on user feedback/complaints, particularly in remote locations. This can lead to faster turnaround times for network troubleshooting [253], and therefore may lower outage periods for users heavily dependent on video streaming. The last three chapters have focused solely on the performance of mobile networks. In the next and final chapter, we explore a system built for fixed broadband that enables faster in-network processing of applications using network accelerators.

Chapter 11

AKIDA: Accelerating In-Network, Transient Compute Elasticity using SmartNICs

11.1 Introduction

The proliferation of the Internet of Things (IoT) and the success of rich cloud services have pushed the horizon of a new computing paradigm, edge computing, that requires faster data processing at the network's edge. The edge computing market is expected to reach close to $9 billion by 2025 [254]. As a specific example, the significant factors driving the growth of the IoT in the manufacturing market include growing demand for industrial automation in the manufacturing industry, rising need for centralized monitoring and predictive maintenance of resources, rise in the number of cost-effective and intelligent connected devices and sensors, among others. To keep up with this demand, there has been a shift to serverless frameworks for computation [255, 256]. Serverless frameworks allow IoT applications to be de-

ployed within minutes with the flexibility to reduce or expand operations seamlessly. For instance, serverless functions provide authentication and encryption capabilities on-site instead of uploading the data over a vulnerable network to the cloud. This requires efficient scale up (or down) of edge computing infrastructure for transient spikes in serverless workloads.

However, managing edge computes capacity on the fly, i.e., *transient compute elasticity*, carries specific challenges [257, 258]. First, the expanding edge deployments are time and resource-intensive. A typical solution is to over-allocate resources for possible future demand. However, over-allocation leads to under-utilization for the most part and is economically undesirable. Edge computing requires careful planning of the available resources *in-situ* to achieve its primary objective of faster processing and reduced latencies. Second, and most importantly, sudden spikes in demand for processing could create compute bottlenecks, leading to service level agreement (SLA) violations. SLA comprises the agreed-upon QoS (Quality of Service) attributes monitored regularly; failing to meet the QoS attributes can attract hefty penalties. In this context, we ask the following research question: *How could we design an architecture that can handle sudden spikes in demand, address transient elasticity, and allocate compute resources efficiently?*

We propose AKIDA, a new edge computing platform that leverages heterogeneous-computing nodes (including domain-specific accelerators like SmartNICs) to dynamically allocate computation requirements for workload spikes with minimal cold start latency. We use SoC-based SmartNICs to predict and intelligently load-balance containerized serverless workloads across the heterogeneous-compute resources. AKIDAuses untapped general-purpose compute on SmartNICs for in-network application processing when demand escalation is imminent. SmartNICs are ideal candidates for application offload because: (i) they are closer to the data ingress pipeline

that enables them to bypass the network stack overhead at the host server, (ii) of the availability and proximity of SoC-based onboard compute for application processing [259, 260], (iii) they are a feasible alternative to the traditional servers for short term compute, and (iv) unused compute cycles on the SmartNICs can be re-purposed for workloads. This is the first study to propose containerized application offload to SoC-based SmartNICs to our knowledge. Although prior works have studied the applicability of offloading specific parts of applications, e.g., using P4 programmability, actor-programming paradigm, etc. [261, 262, 263], those studies are limited to particular applications and require code modification for other types of application offload. In contrast, AKIDAis designed to offload a network of containers onto the SmartNIC, making it truly application-agnostic and scalable. Our platform has three unique elements: (i) a workload predictor, (ii) a traffic distributor, and (iii) an orchestrator. The workload predictor estimates the potential change in demand for the next time horizon by extracting fine-grained input features from historical time-series data. The traffic distributor distributes the traffic based on the transient spikes and CPU load on each cluster node. Finally, the orchestrator sets the threshold levels for intelligent traffic distribution to cluster nodes and manages the end-to-pipeline for application processing. It also can reallocate workloads on the fly to the SmartNICs, if the incoming requests for an application suddenly change.

AKIDA's orchestrator can be generalized for scaling edge across multiple servers and different kinds of SmartNICs. Stated otherwise, our system can be scaled to offload applications across different dimensions of heterogeneity (for instance, if the cluster introduces additional compute nodes). This approach enables us to secure a competitive advantage compared to legacy edge architectures and deployments.

This chapter makes the following key contributions:

- Design of a novel architecture that leverages heterogeneous computing nodes

(SmartNICs and host server) to facilitate efficient handling of transient spikes at the edge;

- Development and characterization of workload predictor and orchestrator that work in tandem to reduce SLA violations, efficiently handle spikes in demand, and reduce cold start latency;

- Characterization of competitive advantages of our architecture through an in-depth analysis of capital expense costs and overhead savings from minimizing SLA violations. Our investigation reveals that capital expenditure (CAPEX) can be reduced by $\geq 1.5\times$, while the operational expenditure (OPEX) can be decreased by $3.5\times$. In addition, our architecture demonstrably reduces SLA violation by as much as 20% in real-world deployments.

11.2 Background

This section provides an overview of multicore SoC-based SmartNICs, and how they are integrated into the edge computing platform. In addition, we briefly discuss the edge computing architecture and explore some common SLA violations typically prevalent in this context.

11.2.1 SmartNICs

There are broadly three categories of network accelerators or SmartNICS: ASIC, FPGA, and SoC-based SmartNICs [264, 262]. In this study, we focus on SoC-based SmartNICs only. Multicore SoC-based SmartNICs use embedded CPU cores to process packets, trading some performance to provide substantially better programmability than ASIC-based designs. (e.g., DPDK-style code can be directly run on a

familiar Linux environment). For instance, Mellanox Bluefield [259] uses general-purpose CPU cores (ARM64), while others, like Netronome [265], have specific cores for network processing.

SoC-based SmartNICs (e.g., Mellanox) have two modes of operation: *Embedded*, and *Separated* modes. The interfaces are mapped to the host OS network stack in embedded mode, and the kernel routes packets from the host. The host OS and the SmartNIC have separate, independent network stacks to process packets in the separated mode. While we observe slightly better tail-latencies from packet processing in embedded mode, the offset from separate mode is negligible. For AKIDA, we adopt the separated mode due to its programmable flexibility and the ability to run containers directly on the SmartNIC's ARM64 OS.

11.2.2 Edge Computing

The adaption of cloud computing platforms is increasing rapidly. However, efficient processing of the data that has been produced at the edge of the network is a challenging task. Data-driven applications are increasingly deployed at the edge and will consequently benefit from edge computing, which we explore here.

Networking bottlenecks:

Compared to the fast-developing cloud-based processing speed, the network bandwidth has reached a standstill. With the growing quantity of data generated at the edge, the rate of data transportation is becoming the bottleneck for the cloud-based computing paradigm. For instance, we expect autonomous vehicles to output a vast amount of data per hour that needs real-time processing. In this instance, edge computing is beneficial over cloud computing because of the significant savings in latency overheads. Additionally, scaling these pipelines for multiple vehicles would require

computation at the edge, not the cloud.

Explosion of IoT: Almost all kinds of electrical devices will become part of IoT, and they will play the role of data producers and consumers, such as air quality sensors, LED bars, streetlights, and even an Internet-connected microwave oven. Reports suggest that the number of IoT devices at the edge will develop to more than billions in a few years [266]. Thus, the raw data produced will be enormous, making conventional cloud computing not efficient enough to handle all this data – application processing at the edge could account for this surge in demand.

Data producers: In the cloud computing paradigm, the end devices at the edge typically are data consumers. For example, they are consuming on-demand video streams on a smartphone. However, vast amounts of data are now produced *by the said-consumers*. Changing from a data consumer to a data producer requires more placement of functionalities at the edge.

11.2.3 SLA Violations

Service Level Agreements are critical when applications are deployed in a Service Oriented Architecture (SOA). SLAs are commonly adopted in cloud computing and, more recently, at the Edge. SLA defines the level of service the consumer expects based on metrics that the application provider lays out. SLA composes of the metrics by which the service is measured, such as monitoring the QoS (Quality of Service) attributes [267, 268], and the remedies or penalties if the metric measurement does not meet the agreed-on service level termed as *SLA Violation*. Some of the most common QoS attributes that are part of SLA are response time and throughput,

we primarily focus on response time.

In Edge Computing, where there are limited resources when the application re-

ceives multiple queries at scale, the response time suffers high tail latency. This problem is further strained when the host OS has an additional background workload for other applications or maintains the edge infrastructure for its Network and Storage needs. This leads to SLA violation and the consumer's poor application Quality of Experience (QoE). We use the response time metric in Sec. 11.4 to evaluate the penalty with and without additional processing units such as SmartNICs.

11.2.4 Need for Accelerators

There has been a lot of research recently in the industry regarding using SmartNICs in cloud data center servers to boost performance by offloading computation in servers by performing network datapath processing. This section explains why SmartNICs are essential in the new generation of high-performance computing servers.

The cost of building an interconnection network for a large cluster can significantly affect the choice of design decisions. With increasing network interface bandwidths, the gap between the network performance and compute performance is widening. This has resulted in increased adoption and deployment of SmartNICs. If SmartNICs were leveraged to offload only network functionalities, it would add 30% more computational capacity to the current servers [269]. Typically, SoC-based SmartNICs are priced at 25-30% the cost of Data Center Servers. Therefore, adding a SmartNIC to perform only network functions is a wise decision. However, the SmartNICs can do more than network functions. As per our initial analysis, the compute capacity of an SoC-based SmartNIC is generally around 40-50% of server compute capacity. If additional compute is required within this range, exploiting the total capacity of SmartNICs to manage workload spikes instead of servers is a

more economical decision. However, all that compute available on SmartNICs is currently primarily used for offloading network functions and services. In most cases, that is a severe under-utilization of the available compute power on SmartNICs. It is this under-utilized compute that AKIDA aims to harvest and make available to the applications.

11.3 System Overview

We begin by providing an overview of AKIDA, an intelligent fabric software framework that can be deployed on any container orchestration supporting Operating Systems such as Servers/Server Racks, Network Switches, or Edge-systems. Figure 11.1 shows the various components of AKIDA framework. The server can host any number of SmartNICs as the number of PCIe buses available. We use Kubernetes as the container orchestration system that runs on the host and SmartNIC OS, and this specialized architecture works only on SoC-based SmartNIC architecture [259].

The major components of our core solution consist of (i) a *traffic distributor* module that distributes the traffic based on the service time and CPU load of each server and SmartNIC, (ii) a *workload prediction* module that uses the history of the workload in a window to predict the workload spikes and (iii) the *AKIDA orchestrator* module manages the workload spikes based on the load on the servers and SmartNICs. In the following, we describe our solution to each module.

11.3.1 Traffic Distributor

The current serverless computing design assumes that all computing resource

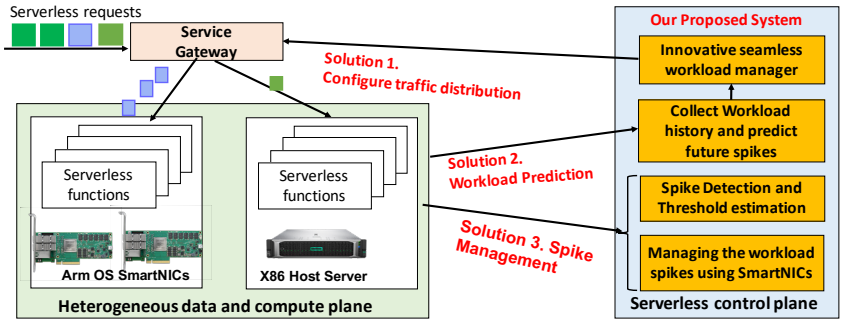


Figure 11.1: System overview.

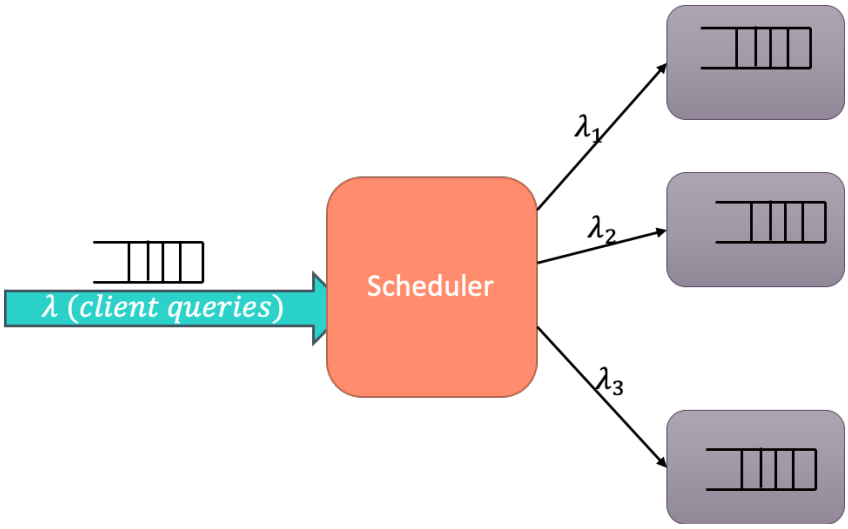


Figure 11.2: Traffic distributor.

nodes are homogeneous and have the same service time and the same amount of load. In this chapter, we show that this assumption leads to degraded performance of workloads running on multiple nodes, especially when one of the compute nodes get overloaded or takes more time to service the requests. To clarify the problem, consider two serverless functions A and B that take 2/10 seconds to run on the SmartNIC and 1/5 second to run on the host OS, respectively, but when the load on the host OS gets overloaded with other workloads, the response times on the host OS changes to 3/8 for functions A and B respectively [1]. In this example, it is better to run the

[1]We note that these numbers are subject to change time to time depending on the workload burst

two functions on the host OS when the host OS is not overloaded, and when it gets overloaded, function A can be offloaded to the SmartNIC.

11.3.2 Traffic Distributor

In our design, the queries first arrive at the API gateway of the scheduler within the SmartNIC OS, where our traffic distributor distributes the traffic according to the service time of each SmartNICs' ARM core or host OS's core within a server. We note that the service time of each function is subject to change depending on the workload spikes. Assuming the requests arrive with the arrival rate of λ and assuming each host OS and SmartNIC have a service rate of μ_i and have an $M/M/1$ queue at each server, the optimal traffic distributor that makes the sojourn time equal for each queue is as follows:

$$\lambda_1 - \mu_1 = \lambda_2 - \mu_2 = ... = \lambda_n - \mu_n \tag{11.1}$$

In other words, the optimal traffic distribution on N servers is as follows:

$$\lambda_i = \mu_i + \frac{\lambda - \sum_{j=1}^{N} \mu_j}{N} \quad i = 1, ..., N \tag{11.2}$$

In the evaluation, we use a heuristic approach and try to avoid distributing the traffic on a cluster node with very high service time due to workload spikes. The queries are then redirected to the appropriate containerized application pods running either on the Host or SmartNIC OS.

and resource congestion on the SmartNICs and host OS servers.

11.3.3 Workload Prediction

To provision the workload spikes proactively to meet the required Service Level Agreement (SLAs), we predict the future workload demands ahead of time. We propose a support vector regression (SVR) prediction model that predicts the workload bursts to trigger the traffic distribution module and also mitigate the impact of containers' cold start latency [270, 271, 272, 273] that can generally lead to a longer response time to application queries otherwise. Our prediction model is based on the past observations of the workload over a window size of W time units. We change the window size dynamically based on the workload variations over time. We increase the training window size if the workload variation over the current window is less than 10% and decreases once the workload variation is more than 20%.

11.3.4 AKIDA's Orchestrator

AKIDA, consists of a resource monitoring module and exploits the output of the prediction module. The resource monitoring module periodically monitors each node's CPU, memory utilization, and service rates in the serverless platform. If the CPU utilization gets higher than a specified threshold Δ, or if the service rate of application X on one of the nodes in the cluster gets higher than the specified SLA, we re-distribute the workload to dampen the spikes. We use the output of the workload prediction module to predict future spikes ahead of time and perform proactive spike management. Pro-active spike management that exploits the prediction module has two benefits: (i) first, we can re-distribute the traffic based on the predicted future workload, which avoids specific server nodes from getting congested, and (ii) second, it mitigates the containers' cold start latency by starting new containers before the actual load arrives. The spike management module updates the service rate, μ_i of each

node in the cluster and requests arrival rates in the traffic distributor module, and triggers a new traffic distribution command if the spikes are higher than a specified threshold or the mean service rate of a node in the cluster increases and violate the specified SLA metric.

11.3.5 Auto-scaler

After splitting the traffic between multiple queues, we scale up/down the number of replicas at each queue. Our auto-scaling algorithm is based on the arrival rate of the predicted workload at time t, (i.e., λ_t), the current number of replicas r_t, and the current service rate of the replicas at each server/SmartNIC (μ_t). We can draw the system utilization as follows:

$$\rho_t = \frac{\lambda_t}{r_t \mu_t} \tag{11.3}$$

Then we calculate the probability that the queue is idle as follows:

$$P_o = 1/[\sum_{n=0}^{r_t-1} \frac{(r_t\rho_t)^n}{n!} + \frac{(r_t\rho_t)_t^r}{r_t!(1-\rho_t}] \tag{11.4}$$

The queue length is

$$L_q = \frac{r_t^{r_t}\rho_t^{r_t+1}}{r_t!(1-rho_t)^2}P_0 \tag{11.5}$$

and the expected waiting time on the queue is $T_q = L_q/\lambda_t$. Given the current number of replicas and the system's service time, we calculate the system's latency $T_q + T_s + 2\delta$ (where 2δ accounts for the auto-scaling startup latency) if the latency was larger than the target SLA, we increment the number of replicas and calculate the optimal number of replicas using a binary search algorithm. If $T_q + T_s + \delta$ was smaller than the target SLA latency, we scale down and find the optimal number of replicas using a binary search algorithm.

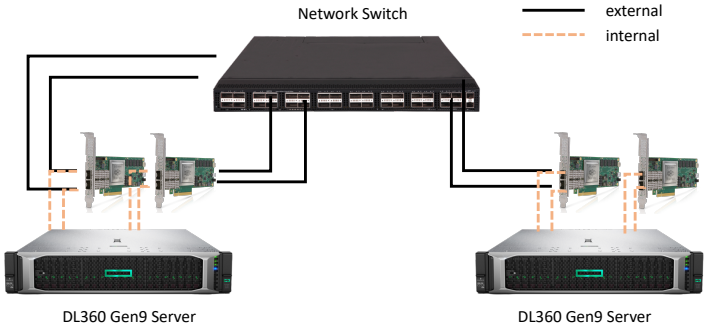


Figure 11.3: Real world experimental setup.

11.4 Competitive Advantages

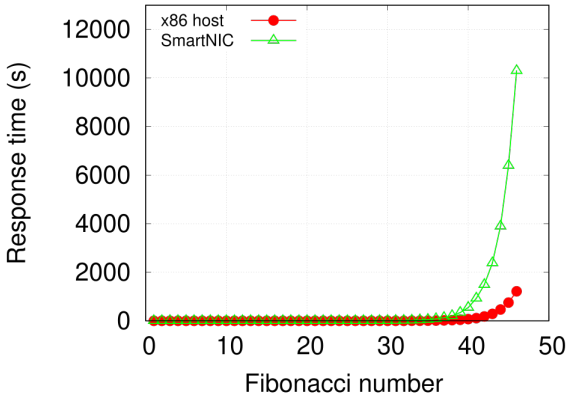
We set up the testbed of AKIDA using DL380 Gen9 Servers and two Mellanox Bluefield [259] SmartNICs per server as shown in Figure 11.4. We deployed a Kubernetes cluster over both server and SmartNIC OS to obtain heterogeneous multicore cluster nodes.

We implemented a prototype based on OpenFaaS serverless infrastructure. We evaluated it on three popular serverless workloads, (i) CPU-intensive Fibonacci function, (ii) latency-sensitive key-value store, and (iii) a sentiment analysis function that uses machine learning to perform natural language processing. We build the functions to run on a multi-architecture platform, including x86 host OS and the SmartNICs' ARM core.

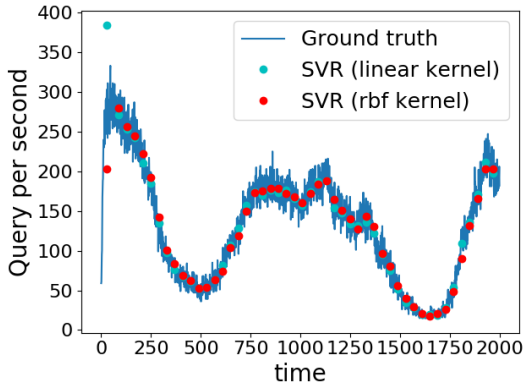
We first run initial experiments to find the compute capacity of SmartNICs by running Fibonacci functions on SmartNICs and Host. We observe the compute capacity close to that of the host's resources. Figure 11.4(a) shows the execution time of running the Fibonacci function on the host OS and the SmartNIC as we increase the Fibonacci number to compute. We observe that SmartNICs have comparable compute capacity as x86-64 Hosts, which assures that the SmartNICs are capable of

running workloads and processing incoming packet traffic.

We also ran initial experiments on an online prediction model to predict future workloads ahead of time to narrow down the best-performing algorithm that works well with our solution. We used 10,000 data points from real serverless workloads that provide an appropriate workload for a ride-sharing application to request a ride [274, 275]. Figure 11.4 shows the workload prediction using the RBF and linear kernel in the SVR prediction model when we train the model over a window size of 100 seconds and predict the future workload d seconds ahead of time. As shown, the RBF kernel performs better than the linear kernel. In the following sub-sections, we investigate data centers' different design choices to manage the load spikes.



a. Response time of the SmartNIC and host OS.

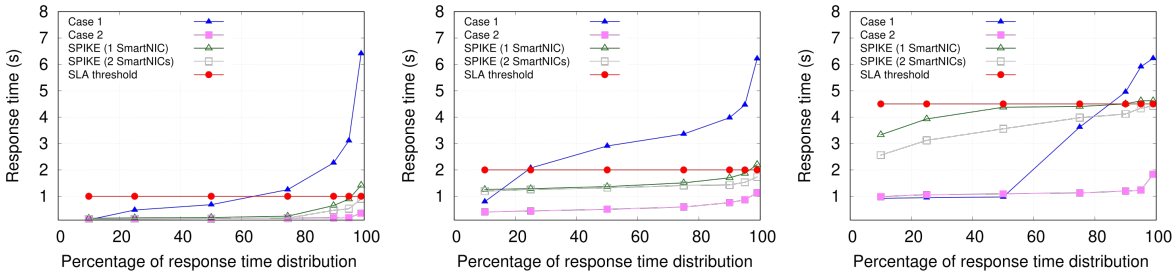


b. Predicting the workload d seconds ahead of time where $d = 10$.

Figure 11.4: Experimental results on the real world testbed.

11.4.1 Performance Benefits

To evaluate the performance benefit of using SmartNICs in the cluster when having a high CPU load, we perform a set of experiments on the three serverless functions in our testbed using OpenFaas serverless platform with the ***hey*** HTTP(S) load generator [276] and emulate transient spikes using a stress tool[277]. Figure 11.5



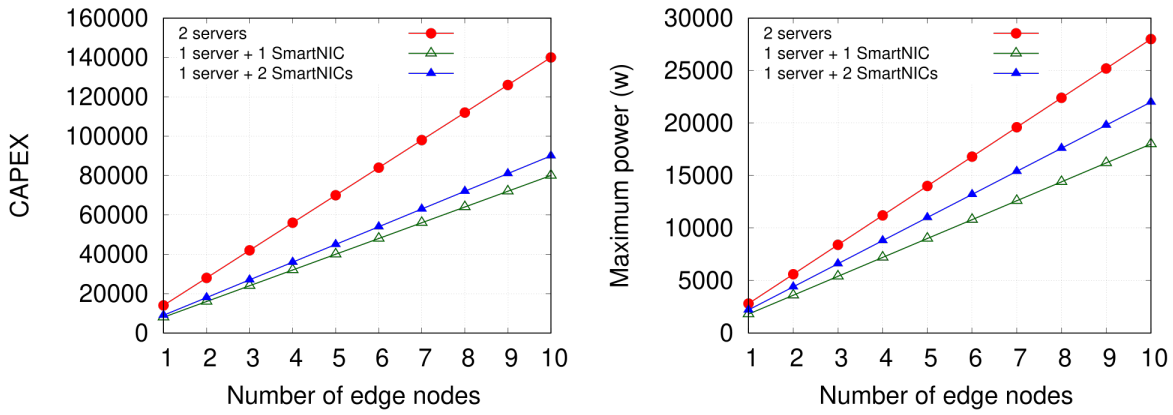
a. Fibonacci. b. Key-value store. c. Sentiment analysis.

Figure 11.5: Response time distribution of different functions.

shows the response time distribution for different functions. The SLA threshold is specified by the application and exposed to the scheduler. We first run the default OpenFaas scheduler on one server, we introduce stress on the host server and increase the average CPU utilization to 80% by running background serverless workload with 200 average queries per second (*Case 1*: 1 server with background workload). The tail latency increases when the host OS has a high load, leading to SLA violations. Adding another server with uniform traffic distribution (default Kubernetes scheduler) in the baseline (2 servers, one with background workload and one without background workload) does not solve the problem since half of the queries are routed to the overloaded host. Next, we run the workload on 2 servers with load-aware proportional traffic distribution (*Case 2*: two servers with proportional traffic distribution similar to AKIDA's traffic distributor). In AKIDA, we detect the overloaded node in the cluster and avoid routing the traffic to that node. We run AKIDA in two cases when having one SmartNIC and two SmarttNICs on the same server. Although the SmartNICs have lower computational power than the host OS when a transient spike overloads the CPU, AKIDA leverages SmartNIC's compute capacity to reduce SLA violations.

11.4.2 Cost Benefits

In this section, we perform a cost analysis of the cluster design choices based on the actual CPU utilization dataset in [278] to compare the network design of over-provisioning the servers to meet SLA during the workload spikes and by using the SmartNICs to manage the spikes. We assume a SmartNIC is about 15-20% of the cost of a server. We calculate CAPEX and OPEX for three resource deployment scenarios, i) two servers, ii) one server and one SmartNIC, and iii) one server and two SmartNICs at each edge node to accommodate the spikes. The x-axis shows the number of edge nodes at each case (i.e 1 edge node + 1 extra server, 1 edge node + 1 SmartNIC, and 1 edge node + 2 SmartNICs). Figure 11.6(a) shows the capital expenses for building a cluster in (i), (ii), and (iii). As shown, the SmartNICs provide an extra computational capacity to the cluster at a much lower cost. The total cost of the cluster reduces by a factor of 1.5 and 1.55 when using one or two SmartNICs at each x86 host, respectively. This section's CAPEX and OPEX cost calculations are based on rough numbers available for cost and maximum energy consumption of the servers and SmartNICs in our testbed. Figure 11.6(b) shows operational expenses by tracking maximum power (one of the main contributors to OPEX) used in the cluster for Cases i, ii, and iii. The SmartNICs used in our testbed are 3.5x more energy efficient than the host server. Figure 11.6(b) shows that the maximum power usage of the cluster reduces by a factor of 1.5 and 1.27 when having one or two SmartNICs at each server, respectively.



a. Capital expenditure. b. Operational expenditure.

Figure 11.6: Operational performance as the cluster size increases.

11.5 Discussion

Industry Implications

There has been a tremendous rise in the adoption of Function-as-as-Service (FaaS), potentially due to their economically attractive services with drastically reduced operational costs compared to Infrastructure-as-a-Service (IaaS). Enterprises like ClearBlade, EdgeConneX, and Edge Intelligence focus on running FaaS workloads at the edge instead of processing data at a data warehouse. We believe AKIDA could drastically improve the scalability of computing resources at a fraction of the cost of adding bare-metal servers, as is traditionally done. In addition, conventional cloud service providers like AWS and Microsoft Azure could benefit from freed-up compute cycles from the host servers (offloaded to SmartNICs), which then can be re-purposed to client-based compute allocation, thereby increasing revenue.

Challenges in Hardware-Offload

In this section, we enumerate some challenges we faced and lessons learned while developing AKIDA. Challenges can broadly be classified into hardware- or software-based, as described below.

Hardware-based: Manufacturers have yet to make SmartNICs universally compatible with all available bare-metal options. Additionally, to scale a platform like AKIDA, it is imperative to have a vendor-agnostic solution. For instance, SoC-based SmartNICs could offer a universal protocol to set up Linux environments effortlessly; they are currently tightly coupled into their vendor-designed architectures, which involves procedural effort for initial setup. However, we note that this is relatively easier to achieve than setting up bare-metal servers that require infrastructure-based protocols to be followed during setup.

SoC-based SmartNICs, which are generally "off-path" in architecture design, contain a "nic-switch" [262] that determines the packet's path once it enters the physical interface. This "nic-switch" has a rigid design based on the vendor's architecture, and therefore, latency performance is skewed for embedded or separated modes. Moreover, most SmartNIC vendors only enable embedded mode, which further challenges the deployment of AKIDA framework—allowing flexibility of the "nic-switch" can potentially lower latency and improve the performance of AKIDA.

Software-based: SoC-based SmartNICs typically embed an ARM64 architecture, while bare-metal solutions usually are built on an x86 architecture. This heterogeneity in system architecture requires us to develop and make available serverless applications (container images) for each system type (ARM65 and x86) – leading to higher temporal overheads. Further, AKIDA framework is developed for SoC-based SmartNICs [264, 262], given the need for an Linux-based operating system to run

container orchestration (that executes the workloads). Further research is required to develop AKIDA-like solutions for ASIC- and FPGA-based SmartNICs.

Previous work [262] has demonstrated the efficiency gain in offloading networking functions onto the SmartNICs. However, SmartNICs have limited computation capacity when compared to server OS. For example, SmartNIC in our testbed has 16 processing threads with 312.50 flops, and the host server has 40 processing threads with 4604.74 available flops. So, modules (in kernel space) that prioritize networking functions over application workloads are necessary to ensure proper functionalities of network processes. Moreover, careful monitoring is needed to ensure the area under the curve (AUC) for workload spikes does not exceed the compute capacity of SmartNICs (serverless functions typically have smaller footprints). Lastly, to minimize cold-start latency of the initial offload (potentially have higher tail latencies), seed pods need to be warmed up on the offload devices at all times. This approach can incur a small computation overhead.

11.6 Related Work

Recent research on SmartNICs focuses on leveraging the compute power to offload various application workloads [261, 263, 262, 279]. The study focuses on moving small functions onto the SmartNIC OS. At the same time, AKIDA is a framework that can offload complete workload containers onto the SmartNIC OS, leveraging the separated-host mode (Sec 11.2.1) of SoC-based SmartNICs. Most other research on SmartNIC hardware offloading is limited to network functions such as load-balancing, firewall, etc. [280]. In AKIDA, we differentiate by offloading application workloads as containers, not just functions. We showcase the benefit and also explore the challenges.

Traffic spike management is a well-explored problem space in the cloud and networks [281, 258], where user queries surge can lead to downtime and poor QoE. At the edge, it can be pretty challenging to deal with traffic spikes, and the most common business solution is over-provisioning computing resources [282, 283]. We differentiate by utilizing SmartNICs to address transient spikes at the edge, thereby enabling transient elasticity of resources. SmartNICs are limited in compute capacity, so it is critical to offload essential workload. Therefore, AKIDA's approach to offload only during traffic spikes validates the approach given the CAPEX and OPEX savings without much application performance degradation.

In that aspect, Serverless applications are gaining popularity to be deployed at the edge for AI, security, and storage workloads [255, 256]. Certain body of research have explored offloading functions of applications onto SmartNICs processing units[261, 262, 263]. For instance, Lambda-NIC [261] demonstrates offloading data plane programming functionality of serverless applications to ASIC SmartNICs. While iPipe [262] offloads applications designed in actor-programming model. In AKIDA, we adopt a novel approach of offloading the entire containerized serverless application (small function containers) onto the SoC-based SmartNIC OS by establishing SmartNIC OS as nodes in the cluster network. While we move the whole container to the Smartnic, [261] rely on P4 programmability to offload a small part of applications to the SmartNic. In [261] the host and the Smartnic are one single node in the Kubernetes cluster, and changing the application requires code modification to offload to the SmartNic, while in our framework, the SmartNic is one of the nodes in the cluster and can leverage the Kubernetes orchestration system for scheduling, auto-scaling, etc.

11.7 Conclusion

This chapter proposes a new platform that leverages the SmartNICs' computational capacity to offload and accelerate serverless workload in the presence of transient traffic spikes at a lower cost. Our solution is three-fold. First, we propose a novel system architecture leveraging container orchestration systems to distribute the workloads between Hosts and SmartNICs based on the demand for transient elasticity of resources. The next challenge we solve is to manage the workload spikes by exploiting the unused computational capacity of the SmartNICs to avoid SLA violations. Finally, we propose a novel workload prediction approach that predicts the transient spikes and starts the containers before the actual load arrives in the system to mitigate the containers' cold start latency. Accounting for transient elasticity using SmartNICs has the added benefit of provisioning a hybrid cloud and edge deployment, with the flexibility to scale edge deployments when required. This could lead to faster turnaround times for system administrators executing decisions to allocate compute cycles. While this study focuses on transient elasticity for workload spikes, AKIDA architecture can be leveraged for building a generalized system for federated edge infrastructure with heterogeneous DPUs such as GPU and SmartNICs.

Part III

Conclusion

Chapter 12

Conclusion and Summary

12.1 Conclusion

Access to the Internet is critical to the information capital of all communities. Connectivity is a central component in modern-day technology; however, current designs and their deployments for widespread connectivity are driven by economic feasibility. Such deployment policies have an adverse effect on the availability of usable service quality across geographies. Even in cases where networks are well-provisioned, they are done based on a median historical demand. Stated otherwise, network planning and deployment are still *reactive* in nature. However, we have witnessed massive socio-economic changes over relatively short periods, a phenomenon well exemplified by the COVID-19 pandemic and slated to continue in the future. To ensure basic connectivity and usable service quality in rapidly changing times, network planning and deployment strategies need to be more *predictive*, not just proactive.

This dissertation integrates the field of network performance measurement with predictive systems. It focuses on using passive and inexpensively-accessible active measurements to inform the deployment of adaptive systems in real-world networks.

The presented research approach has two cornerstones: (i) identification of problems in access and adoption of technology through in-depth measurements, and (ii) design and implementation of predictive network systems to solve the identified problems. Some of the fundamental problems in this domain are in core network systems design and implementation, while others are for studying specific environmental factors that influence network design, performance, and utilization. The contributions of our research to date lie in the intersection of these problem sets and have the following primary dimensions:

1. **Network characterization** of real-world networks in order to fully understand the inadequacies users currently face [35, 4, 84, 97]

2. **Predictive system design** that are informed by our analysis to enhance the performance of current network deployments [90, 109, 5]

Our work demonstrates multiple opportunities to rethink current designs and implement demand-aware deployment strategies. We have characterized last-mile connectivity in numerous locations to gain insight into performance and built systems based on our observations. We then designed and implemented systems that can infer congestion on the network side and systems that accurately predict the quality of experience for the end-users.

12.1.1 Network characterization

To increase connectivity, we must first understand the performance of networks as they operate today. We study both mobile- as well as fixed-broadband networks so that we may identify avenues for improvement.

Mobile devices today are equipped with multiple interfaces, creating usage opportunities for protocols such as Multipath TCP (MPTCP), which enable devices to

use more than one interface concurrently. One of the biggest hurdles in implementing MPTCP is the heterogeneity in performance characteristics across multiple interfaces. We perform controlled and real-world experiments over multiple paths with differing loss rates and round trip latencies to assess the effect of primary path selection and the range of issues that arise from selecting the under-performing path. Using results from our experiments, we show how heterogeneous paths can adversely affect MPTCP performance, especially when one path is lossy.

In the context of FCC LTE maps that drew intense criticism for overstating coverage, our assessment in New Mexico indicates that physical assessments of cellular coverage and quality are essential for evaluating actual user experience. However, measurement campaigns can be resource, time, and labor-intensive; therefore, more scalable measurement strategies are urgently needed. To that end, we assess the accuracy of a low-cost, small form-factor RTL-SDR for sensing LTE eNodeB signal strength over a wide area through integration with an off-the-shelf quadcopter UAS. Our observations reveal that the simple RTL-SDR has comparable accuracy to expensive solutions and can estimate quality within one gradation of accuracy compared to user equipment (i.e., mobile phones).

Next, we evaluate the validity of the criticism towards FCC coverage maps using a quantitative analysis of both the dataset from which the FCC based its report, a crowd-sourced LTE coverage dataset, and our measurements on the ground. To that end, we performed a controlled measurement campaign in northern New Mexico in May 2019. We find that both FCC and crowd-sourced datasets report higher coverage relative to our controlled measurements, with the former showing a higher degree of over-reporting than the latter. Understanding the causes of these inconsistencies is vital for effectively using crowd-sourced data to measure LTE coverage, especially as crowd-sourcing is increasingly viewed as preferable to provider reports.

Finally, coverage reports that show service quality as a binary indicator does not cleanly capture end-user experience. We develop a robust measurement suite to conduct a unique measurement campaign in tribal, rural, and urban regions, representing a variety of under-provisioned, congested, and well-provisioned operational LTE networks run by four major providers. In the regions we study, LTE networks in tribal and rural locations have significantly higher QoE degradation than urban locations, typically also worse than congested networks.

Contributions and impact: Our work has gained attention in the networking research community. Our MPTCP study was presented at ICCCN 2019 conference. The observations in our UAS assessment of LTE networks were presented at IEEE MASS 2019. The comparison of FCC coverage maps with our ground truth measurement was published in the Communications of the ACM (March 2022). Lastly, our *Coverage is not binary* paper was published at ICCCN 2021.

12.1.2 Predictive system design

We leverage our insights from the measurement studies to design predictive systems aimed at closing the gap between connectivity and actual usability.

To remedy this disparity between reported coverage and actual service quality, we design and implement Lumos – a novel solution to infer overload in LTE networks based on messages broadcast by the eNodeB. Through the analysis of multiple message types, we draw clear comparisons between instances of high network utilization and typical operating conditions for several eNodeBs. Our results indicate that eNodeBs demonstrate measurable performance differences indicative of overload conditions.

It is equally critical to draw parallels between network overload and congestion.

We design Edain – a comprehensive networking monitoring suite, to evaluate QoS and QoE metrics due to network overload. Our system implementation shows that eNodeBs demonstrate measurable performance differences indicative of overload conditions and network congestion.

Further, the wireless nature of LTE networks necessitates that QoE is evaluated in multiple locations per base station as factors such as signal availability may have significant spatial variation. As we studied the performance across networks that vary spatially and temporally, it became critical to create a system that infers the state of QoE for end-users at scale, using inexpensive and easily accessible measurements. Therefore, we built a comprehensive set of network models that accurately predicts various QoE metrics, such as rebuffering and quality switching events.

Lastly, there has been a rise in demand for processing applications at the edge. However, scaling edge computing poses critical challenges. Service-level agreement (SLA) violations typically occur due to workload bursts. We design and implement a new architecture that strategically harvests the untapped compute capacity of the SmartNICs to offload transient workload spikes, thereby reducing the SLA violations. We propose AKIDA, a low-cost and scalable platform that orchestrates seamless offloading serverless workloads to the SmartNICs, eliminating the need for pre-allocating expensive compute power and over-utilization of host servers.

Contributions and impact: Our work on Lumos was the first to use passive measurements to detect network overload. The study was presented at IMC 2019, one of the most selective network measurement conferences. We extended Lumos to validate network congestion using Edain, and the paper was published in IEEE Transactions on Mobile Computing – a highly regarded journal. Our work on developing predicting systems for QoE measurement was presented at Passive and Active Measurement 2021, a top measurement conference. Finally, our work on AKIDA was the

only project selected among 400 summer intern projects at Hewlett Packard Labs to win the 2020 Best-in-Class award.

12.2 Summary

Access to technology will continue to drive development. Commercial network expansions have successfully brought connectivity to most of the world. Still, the remaining unconnected areas present significant challenges due to the sparseness of users and lack of purchasing power. Providing adequate connectivity in such locations necessitates a thorough grasp of the current difficulties in designing network solutions that deliver efficient service. This dissertation advances the field by analyzing network utilization and performance in previously unexplored scenarios. We use these insights to find creative ways to enhance connectivity in these locations. Current monolithic and homogenous solutions will continue prohibitively expensive and undesirable for a substantial section of the world population. Therefore, we must continue exploring a wide range of networking technologies designed for resilient networks.

Bibliography

[1] Statista, "Global digital population as of April 2022." https://www.statista.com/statistics/617136/digital-population-worldwide/, 2022.

[2] Cisco, "Cisco Visual Networking Index (VNI) Forecast Highlights." https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_Device_Growth_Traffic_Profiles.pdf.

[3] U. S. Congress, "Public Law 111–5 – 111th Congress." https://www.govinfo.gov/content/pkg/PLAW-111publ5/pdf/PLAW-111publ5.pdf.

[4] V. Adarsh, M. Nekrasov, U. Paul, T. Mangla, A. Gupta, M. Vigil-Hayes, E. Zegura, and E. Belding, *Coverage is Not Binary: Quantifying Mobile Broadband Quality in Urban, Rural, and Tribal Contexts*, in *2021 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–9, IEEE, 2021.

[5] V. Adarsh, M. Nekrasov, U. Paul, A. Ermakov, A. Gupta, M. Vigil-Hayes, E. Zegura, and E. Belding, *Too Late for Playback: Estimation of Video Stream Quality in Rural and Urban Contexts*, in *International Conference on Passive and Active Network Measurement*, pp. 141–157, Springer, 2021.

[6] Federal Communications Commission, "Broadband Deployment Report." https://www.fcc.gov/reports-research/reports/broadband-progress-reports/2019-broadband-deployment-report, May, 2019.

[7] Ericsson, "Ericsson Mobility Report Q4 Update February 2022." https://www.ericsson.com/492615/assets/local/reports-papers/mobility-report/documents/emr-quarterly2021-update.pdf, 2022.

[8] International Telecommunication Union, "Last-mile Internet Connectivity: Toolkit." https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/Documents/Events/2021/LMC/LMC%20-ASP%20.pdf, 2021.

[9] Government Accountability Office, *BROADBAND INTERNET: FCC's Data Overstate Access on Tribal Lands*, September, 2018.

[10] Rural Wireless Association, "Challenges Faced by Small Wireless Providers in Measuring LTE Coverage." https://ruralwireless.org/rwa-welcomes-fcc-investigation-into-violation-of-mobility-fund-phase-ii-mapping-rules/, 2019.

[11] FCC, "Communications Status Report for Areas Impacted by Tropical Storm Harvey, September 1, 2017." http://transition.fcc.gov/Daily_Releases/Daily_Business/2017/db0901/DOC-346475A1.pdf, September, 2017.

[12] FCC, "Communications Status Report for Areas Impacted by Hurricane Irma - September 7, 2017." https://transition.fcc.gov/DailyReleases/DailyBusiness/2017/db0907/DOC-346607A1.pdf, September, 2017.

[13] FCC, "Communications Status Report for Areas Impacted by Hurricane Maria - September 21, 2017." https://transition.fcc.gov/Daily_Releases/Daily_Business/2017/db0921/DOC-346840A1.pdf, September, 2017.

[14] P. Schmitt, D. Iland, M. Zheleva, and E. Belding, *HybridCell: Cellular Connectivity on the Fringes with Demand-driven Local Cells*, in *IEEE INFOCOM*, pp. 1–9, IEEE, 2016.

[15] Southern California Tribal Chairmen's Association, "Southern California Tribal Digital Village." https://sctca.net/southern-california-tribal-digital-village/.

[16] United Nations, "Resolution 32/13: The promotion, protection and enjoyment of human rights on the Internet." https://http://undocs.org/A/HRC/RES/32/13, July, 2016.

[17] ITU, "ICT Facts and Figures: The World in 2016." https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf, May, 2016.

[18] ITU, "ICT Facts and Figures: The World in 2017." https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2017.pdf, May, 2017.

[19] K. Salemink, D. Strijker, and G. Bosworth, *Rural development in the digital age: A systematic literature review on unequal ICT availability, adoption, and use in rural areas*, *Journal of Rural Studies* **54** (2017) 360–371.

[20] U. Paul, J. Liu, D. Farias-llerenas, V. Adarsh, A. Gupta, and E. Belding, *Characterizing Internet Access and Quality Inequities in California M-Lab Measurements*, in *Proceedings of the ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS '22)*, 2022.

[21] U. Paul, J. Liu, V. Adarsh, M. Gu, A. Gupta, and E. Belding, *Characterizing performance inequity across us ookla speedtest users*, *arXiv preprint arXiv:2110.12038* (2021).

[22] J. A. Van Dijk, *Digital Divide Research, Achievements and Shortcomings*, *Poetics* **34** (2006), no. 4-5 221–235.

[23] P. DiMaggio, E. Hargittai, *et. al.*, *From the 'Digital Divide'to 'Digital Inequality': Studying Internet use as Penetration increases*, *Princeton: Center for Arts and Cultural Policy Studies, Woodrow Wilson School, Princeton University* **4** (2001), no. 1 4–2.

[24] Federal Communications Commission, "Broadband Deployment Report." https://www.fcc.gov/reports-research/reports/broadband-progress-reports/2018-broadband-deployment-report, February, 2018.

[25] Grant Samms (Forbes), "As Cities Face COVID-19, The Digital Divide Becomes More Acute." https://www.forbes.com/sites/pikeresearch/2020/04/02/as-cities-face-covid-19-the-digital-divide-becomes-more-acute/#277c93e558c5, April, 2020. (Accessed on 05/10/2020).

[26] Amanda Holpuch (The Guardian), "US's Digital Divide 'is going to kill people' as COVID-19 exposes inequalities." https://www.theguardian.com/world/2020/apr/13/coronavirus-covid-19-exposes-cracks-us-digital-divide, April, 2020. (Accessed on 05/10/2020).

[27] Klint Finley (The Wired), "When School is Online, the Digital Divide Grows Greater." https://www.wired.com/story/school-online-digital-divide-grows-greater/, Apr, 2020. (Accessed on 05/10/2020).

[28] Federal Communications Commission, "LTE Coverage by Number of Providers - YE 2018." https://www.fcc.gov/reports-research/maps/lte-coverage-number-providers-ye-2018/, 2020.

[29] Federal Communications Commission., "Connect America Fund (CAF)." https://www.fcc.gov/general/connect-america-fund-caf, February, 2017.

[30] J. E. Prieger, "Mobile data roaming and incentives for investment in rural broadband infrastructure." https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3391478, 2017.

[31] J. Kahan, "It's time for a new approach for mapping broadband data to better serve americans." https://blogs.microsoft.com/on-the-issues/2019/04/08/its-time-for-a-new-approach-for-mapping-broadband-data-to-better-serve-americans/, Apr, 2019.

[32] Rural Wireless Association., "RWA Calls for FCC Investigation of T-Mobile Coverage Data." https://ruralwireless.org/rwa-calls-for-fcc-investigation-of-t-mobile-coverage-data/, 2018.

[33] OpenSignal, "Open Signal 3G and 4G LTE Cell Coverage Map." http://opensignal.com, 2016.

[34] Skyhook, "Coverage area." https://www.skyhook.com/coverage-map, 2019.

[35] M. Nekrasov, V. Adarsh, U. Paul, E. Showalter, E. Zegura, M. Vigil-Hayes, and E. Belding, *Evaluating LTE Coverage and Quality from an Unmanned Aircraft System*, in *Proceedings of the IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) 2019*, pp. 171–179.

[36] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, *Tranco: A Research-oriented Top Sites Ranking Hardened against Manipulation*, in *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, Internet Society, 2019.

[37] C. Bureau, "A Comparison of Rural and Urban America: Household Income and Poverty." https://www.census.gov/newsroom/blogs/random-samplings/2016/12/a_comparison_of_rura.html, 2016.

[38] M. Ratcliffe, C. Burd, K. Holder, and A. Fields, *Defining Rural at the US Census Bureau*, *American community survey and geography brief* **1** (2016), no. 8.

[39] SanDiego.org, "St. patricks day parade." http://www.stpatsparade.org/parade-festival-schedule.html, 2019.

[40] S. S. Diego, "Shamrock concert." https://www.sandiegoshamrock.com/, 2019.

[41] SanDiego.org, "Adams Avenue Street Fair." https://www.adamsavenuebusiness.com/event-info/adams-avenue-street-fair/, 2019.

[42] SDparks.org, "Waterfront Park." http://www.sdparks.org/content/sdparks/en/park-pages/Waterfront.html, 2019.

[43] A. Summit, "AI Summit San Francisco." https://sanfrancisco.theaisummit.com/, 2019.

[44] Institute for Policy Research, "What Drives Native American Poverty?." https://www.ipr.northwestern.edu/news/2020/redbird-what-drives-native-american-poverty.html.

[45] Hansi Lo Wang (NPR), "Native Americans On Tribal Land Are 'The Least Connected' To High-Speed Internet." https://www.npr.org/2018/12/06/673364305/native-americans-on-tribal-land-are-the-least-connected-to-high-speed-internet, December, 2018.

[46] Federal Communications Commission, "FCC Mobile Broadband Dataset." https://www.fcc.gov/form-477-mobile-voice-and-broadband-coverage-areas, 2019.

[47] S. Meinrath, H. Bonestroo, G. Bullen, A. Jansen, S. Mansour, C. Mitchell, C. Ritzo, and N. Thieme, "Broadband availability and access in rural pennsylvania." https://www.rural.palegislature.us/broadband/Broadband_Availability_and_Access_in_Rural_Pennsylvania_2019_Report.pdf, June, 2019.

[48] Measurement Lab, "Open Internet Measurement." https://www.measurementlab.net/about/, January, 2019.

[49] FCC, "Understanding wireless telephone coverage." https://fcc.gov/consumers/guides/understanding-wireless-telephone-coverage-areas.

[50] CellMapper, "Crowd-sourced Cellular Tower and Coverage Mapping Service." https://www.cellmapper.net/map, 2019.

[51] C. Hurley, R. Rogers, F. Thornton, and B. Baker, *Wardriving and Wireless Penetration Testing*. Syngress, 2007.

[52] M. C. Batistatos, G. E. Athanasiadou, D. A. Zarbouti, G. V. Tsoulos, and N. C. Sagias, *LTE ground-to-air measurements for UAV-assisted cellular networks*, in *12th European Conference on Antennas and Propagation (EuCAP)*, IET, 2018.

[53] G. E. Athanasiadou, M. C. Batistatos, D. A. Zarbouti, and G. V. Tsoulos, *LTE ground-to-air field measurements in the context of flying relays*, *Wireless Communications* **26** (2019), no. 1 12–17.

[54] C. Desmond, "Verizon Uses Drones During Disasters like Hurricane Harvey." https://www.roboticstomorrow.com/article/2017/11/article/11068, Nov, 2017.

[55] M. Nekrasov, R. Allen, and E. Belding, *Performance Analysis of Aerial Data Collection from Outdoor IoT Sensor Networks using 2.4 GHz 802.15.4*, in *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, ACM, 2019.

[56] M. Nekrasov, R. Allen, I. Artamonova, and E. Belding, *Optimizing 802.15.4 Outdoor IoT Sensor Networks for Aerial Data Collection*, *Sensors* **19** (2019), no. 16.

[57] Osmocom, "RTL-SDR." `https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr`.

[58] F. Minucci, S. Rajendran, B. V. d. Bergh, S. Pollin, D. Giustiniano, H. Cordobés, R. C.-P. Fuchs, V. Lenders, *et. al.*, *Electrosense-spectrum sensing with increased frequency range*, in *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*, Junction Publishing, 2018.

[59] J. A. del Peral-Rosado, J. M. Parro-Jiménez, J. A. López-Salcedo, G. Seco-Granados, P. Crosta, F. Zanier, and M. Crisci, *Comparative results analysis on positioning with real LTE signals and low-cost hardware platforms*, in *7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, IEEE, 2014.

[60] X. Lin, R. Wiren, S. Euler, A. Sadam, H.-L. Maattanen, S. D. Muruganathan, S. Gao, Y.-P. E. Wang, J. Kauppi, Z. Zou, and V. Yajnanarayana, *Mobile networks connected drones: Field trials, simulations, and design insights*, *arXiv* (2018).

[61] O. Genc, M. Bayrak, and E. Yaldiz, *Analysis of the Effects of GSM Bands to the Electromagnetic Pollution in the RF Spectrum*, *Progress In Electromagnetics Research* **101** (2010).

[62] S.-K. Noh and D. Choi, *Propagation Model in Indoor and Outdoor for the LTE Communications*, *International Journal of Antennas and Propagation* (2019).

[63] K. Shamaei, J. Khalife, and Z. M. Kassas, *Exploiting LTE signals for navigation: Theory to implementation*, *Transactions on Wireless Communications* **17** (2018), no. 4 2173–2189.

[64] A. Saeed, K. A. Harras, E. Zegura, and M. Ammar, *Local and low-cost white space detection*, in *37th International Conference on Distributed Computing Systems (ICDCS)*, June, 2017.

[65] O. Simpson and Y. Sun, *LTE RSRP, RSRQ, RSSNR and local topography profile data for RF propagation planning and network optimization in an urban propagation environment*, *Data in brief* **21** (2018) 1724–1737.

[66] C. Lin, K. Sandrasegaran, H. A. M. Ramli, and R. Basukala, *Optimized performance evaluation of LTE hard handover algorithm with average RSRP constraint*, *CoRR* **abs/1105.0234** (2011).

[67] K. Dimou, M. Wang, Y. Yang, M. Kazmi, A. Larmo, J. Pettersson, W. Muller, and Y. Timner, *Handover within 3GPP LTE: Design Principles and Performance*, in *70th Vehicular Tech. Conf.*, IEEE, 2009.

[68] P. Legg, G. Hui, and J. Johansson, *A Simulation Study of LTE Intra-frequency Handover Performance*, in *72nd Vehicular Tech. Conf.*, IEEE, 2010.

[69] D. Aziz and R. Sigle, *Improvement of LTE Handover Performance through Interference Coordination*, in *69th Vehicular Tech. Conf.*, IEEE, 2009.

[70] J. Kurjenniemi, T. Henttonen, and J. Kaikkonen, *Suitability of RSRQ measurement for quality based inter-frequency handover in LTE*, in *Intern. Symposium on Wireless Communication Sys.*, IEEE, 2008.

[71] M. Anas, F. D. Calabrese, P. E. Mogensen, C. Rosa, and K. I. Pedersen, *Performance evaluation of received signal strength based hard handover for UTRAN LTE*, in *65th Vehicular Tech. Conf.*, IEEE, 2007.

[72] J. Kurjenniemi and T. Henttonen, *Effect of measurement bandwidth to the accuracy of inter-frequency RSRP measurements in LTE*, in *19th International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, 2008.

[73] F. Afroz, R. Subramanian, R. Heidary, K. Sandrasegaran, and S. Ahmed, *SINR, RSRP, RSSI and RSRQ measurements in long term evolution networks*, *International Journal of Wireless & Mobile Networks* (2015).

[74] C. Ide, B. Dusza, and C. Wietfeld, *Performance of channel-aware M2M communications based on LTE network measurements*, in *24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, 2013.

[75] C. Ide, R. Falkenberg, D. Kaulbars, and C. Wietfeld, *Empirical analysis of the impact of LTE downlink channel indicators on the uplink connectivity*, in *83rd Vehicular Tech. Conf.*, IEEE, 2016.

[76] C. S. Park and S. Park, *Analysis of RSRP measurement accuracy*, *Communications Letters* **20** (2016), no. 3 430–433.

[77] H. C. Nguyen, R. Amorim, J. Wigard, I. Z. Kovacs, and P. Mogensen, *Using LTE networks for UAV command and control link: A rural-area coverage analysis*, in *86th Vehicular Tech. Conf.*, IEEE, 2017.

[78] R. Amorim, H. Nguyen, P. Mogensen, I. Z. Kovács, J. Wigard, and T. B. Sørensen, *Radio channel modeling for UAV communication over cellular networks*, *Wireless Communications Letters* **6** (2017), no. 4 514–517.

[79] B. Lubek, "Network Monitor." `https://github.com/caarmen/network-monitor`.

[80] Taoglas, "TG.45.8113 Apex III Wideband 5G/4G Dipole Terminal Antenna." `https://cdn3.taoglas.com/datasheets/TG.45.8113.pdf`, 2019.

[81] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, *srsLTE: An Open-Source Platform for LTE Evolution and Experimentation*, in *ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization 2016*, p. 25–32.

[82] SignalBooster, "What is strong signal in dBm for 4G?." `https://www.signalbooster.com/blogs/news/differences-between-3g-1x-vs-4g-lte-signal-strength-in-dbm`, April, 2016.

[83] Pew Research Center, "Mobile Fact Sheet." `https://www.pewresearch.org/internet/fact-sheet/mobile/`, June, 2019.

[84] T. Mangla, E. Showalter, V. Adarsh, K. Jones, M. Vigil-Hayes, E. Belding, and E. Zegura, *A Tale of Three Datasets: Towards Characterizing Mobile Broadband Access in the United States*, *Communications of the ACM* **65** (2022), no. 3 67–74.

[85] Federal Communications Commission, "Coverage Maps Investigation." `https://docs.fcc.gov/public/attachments/DOC-361165A1.pdf`, 2019.

[86] T. Norris, P. L. Vines, and E. M. Hoeffel, *The American Indian and Alaska Native Population: 2010*, United States Census Bureau, February, 2012.

[87] BroadbandNow, "Internet Access in New Mexico." `https://broadbandnow.com/New-Mexico`, 2019.

[88] Sandvine, "Mobile Phenomena Report 2020." `https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2020/Phenomena/Mobile%20Phenomena%20Report%201H%202020%2020200219.pdf`, Feb, 2020.

[89] Statista, "Penetration Rate of Subscription Video-on-demand (SVOD) Services Worldwide in the 1st Quarter of 2019." `https://www.statista.com/statistics/813698/svod-reach-by-country/`, 2019.

[90] V. Adarsh, M. Nekrasov, E. Zegura, and E. Belding, *Packet-Level Overload Estimation in LTE Networks Using Passive Measurements*, in *Proceedings of the Internet Measurement Conference*, IMC '19, 2019.

[91] U. Paul, A. Ermakov, M. Nekrasov, V. Adarsh, and E. Belding, # *Outage: Detecting Power and Communication Outages from Social Networks*, in *Proceedings of The Web Conference 2020*, pp. 1819–1829, 2020.

[92] 3GPP TS 136.214, *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer - Measurements*, April, 2010.

[93] 3GPP TS 36.214 Release 12, *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer Measurements*, June, 2016.

[94] N. Awad and I. Mkwawa, *The Impact Of The Reference Signal Received Power To Quality Of Experience For Video Streaming Over LTE Network*, in *Proceedings of the Annual Conference on New Trends in Information Communications Technology Applications (NTICT)*, 2017.

[95] E. Alimpertis, A. Markopoulou, C. Butts, and K. Psounis, *City-Wide Signal Strength Maps: Prediction with Random Forests*, in *Proceedings of the The Web Conference*, WWW, pp. 2536–2542, 2019.

[96] Kali Tools, "Hping3 Package." `https://tools.kali.org/information-gathering/hping3`, 2019.

[97] V. Adarsh, P. Schmitt, and E. Belding, *MPTCP Performance over Heterogenous Subpaths*, in *Proceedings of the 28th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, IEEE, 2019.

[98] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner, *Assessing Effect Sizes of Influence Factors towards a QoE Model for HTTP Adaptive Streaming*, in *Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 111–116, IEEE, 2014.

[99] I. Sodagar, *The MPEG-DASH Standard for Multimedia Streaming over the Internet*, *IEEE Multimedia* **18** (2011), no. 4 62–67.

[100] A. Bouch, A. Kuchinsky, and N. Bhatti, *Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service*, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 297–304, 2000.

[101] N. Bhatti, A. Bouch, and A. Kuchinsky, *Integrating User-perceived Quality into Web Server Design*, *Computer Networks* **33** (2000), no. 1 1–16.

[102] F. Bronzino, P. Schmitt, S. Ayoubi, G. Martins, R. Teixeira, and N. Feamster, *Inferring Streaming Video Quality from Encrypted Traffic: Practical Models and Deployment Experience*, *Proceedings of the ACM Meas. Anal. Comput. Syst.* (December, 2019).

[103] M. Ghobadi, Y. Cheng, A. Jain, and M. Mathis, *Trickle: Rate Limiting YouTube Video Streaming*, in *Proceedings of the USENIX Annual Technical Conference (ATC)*, pp. 191–196, 2012.

[104] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, *Understanding the Impact of Video Quality on User Engagement*, in *Proceedings of the ACM SIGCOMM Conference*, 2011.

[105] Selenium, "The Selenium Browser Automation Project." https://www.selenium.dev/documentation/en/.

[106] W3C, "Navigation Timing." https://www.w3.org/TR/navigation-timing/#processing-model.

[107] Statista, "Most Popular Video Streaming Services in the US." https://www.statista.com/statistics/910895/us-most-popular-video-streaming-services-by-reach/, 2019.

[108] YouTube, "YouTube Player API Reference for iframe Embeds." https://developers.google.com/youtube/iframe_api_reference, 2019.

[109] V. Adarsh, M. Nekrasov, U. Paul, and E. M. Belding, *Estimation of Congestion from Cellular Walled Gardens using Passive Measurements*, *IEEE Transactions on Mobile Computing* (2021) 1–1.

[110] Zoom, "Minimum System Requirements." https://support.zoom.us/hc/en-us/articles/201362023-System-requirements-for-Windows-macOS-and-Linux.

[111] Federal Communications Commission, "Measuring Broadband America." https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-mobile-2013-2018, February, 2018.

[112] V. Krishnamoorthi, N. Carlsson, E. Halepovic, and E. Petajan, *BUFFEST: Predicting Buffer Conditions and Real-time Requirements of HTTP (S) Adaptive Streaming Clients*, in *Proceedings of the 8th ACM on Multimedia Systems Conference*, pp. 76–87, 2017.

[113] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler, *Waiting Times in Quality of Experience for Web based Services*, in *Fourth International Workshop on Quality of Multimedia Experience*, pp. 86–96, IEEE, 2012.

[114] S. Grondin, *Timing and Time Perception: A Review of Recent Behavioral and Neuroscience Findings and Theoretical Directions*, *Attention, Perception, & Psychophysics* **72** (2010), no. 3 561–582.

[115] R. Schatz, S. Egger, and A. Platzer, *Poor, Good Enough or Even Better? Bridging the Gap between Acceptability and QoE of Mobile Broadband Data Services*, in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2011.

[116] J. Laiho, A. Wacker, and T. Novosad, *Radio Network Planning and Optimisation for UMTS*, vol. 2. Wiley Online Library, 2002.

[117] A. W. T. Tsui, W.-C. Lin, W.-J. Chen, P. Huang, and H.-H. Chu, *Accuracy Performance Analysis Between War Driving and War Walking in Metropolitan Wi-Fi Localization*, *IEEE Transactions on Mobile Computing* **9** (2010), no. 11 1551–1562.

[118] Root Metrics Inc., "Metro RootScore Reports." `http://www.rootmetrics.com/en-US/home`, 2020.

[119] SamKnows, *Methodology and Technical Information Relating to the SamKnows Testing Platform*, *Document Reference: SQ301-002-EN* (2012).

[120] Tutela Inc., " Crowdsourced mobile data." `https://www.tutela.com/`, 2020.

[121] Measurement Lab, "The M-Lab NDT Data Set." `https://measurementlab.net/tests/ndt`, (2009-02-11 – 2015-12-21).

[122] F. Wamser, N. Wehner, M. Seufert, P. Casas, and P. Tran-Gia, *YouTube QoE Monitoring with YoMoApp: A Web-Based Data Interface for Researchers*, in *Proceedings of the Network Traffic Measurement and Analysis Conference*, pp. 1–2, IEEE, 2018.

[123] CAIDA, "CAIDA Data - Overview of Datasets, Monitors, and Reports." `https://www.caida.org/data/overview/`, 2019.

[124] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, *An In-Depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance*, in *Proceedings of the ACM SIGCOMM Conference*, p. 363–374, 2013.

[125] A. Nikravesh, D. R. Choffnes, E. Katz-Bassett, Z. M. Mao, and M. Welsh, *Mobile Network Performance from User Devices: A Longitudinal, Multidimensional Analysis*, in *Proceedings of the Passive and Active Measurement Conference*, 2014.

[126] E. Roberts, D. Beel, L. Philip, and L. Townsend, *Rural resilience in a digital society*, *Journal of Rural Studies* **54** (2017) 355–359.

[127] A. Lutu, D. Perino, M. Bagnulo, E. Frias-Martinez, and J. Khangosstar, *A characterization of the COVID-19 pandemic impact on a mobile network operator traffic*, in *Proceedings of the ACM Internet Measurement Conference*, pp. 19–33, 2020.

[128] National Telecommunications and Information Administration, "NTIA Comments on Modernizing the FCC Form 477." `https://www.ntia.doc.gov/files/ntia/publications/ntia_comments_on_modernizing_the_fcc_form_477_data_program.pdf`, 2019. Accessed: 2021-03-01.

[129] OpenCelliD, "The World's Largest Open Database of Cell Towers." https://opencellid.org/, 2020.

[130] Federal Communications Commission, "FCC Centroid Methodology." https://docs.fcc.gov/public/attachments/DA-16-1107A1_Rcd.pdf, 2019. Accessed: 2021-03-01.

[131] D. Major, R. Teixeira, and J. Mayer, *No wan's land: Mapping us broadband coverage with millions of address queries to isps*, in *Proceedings of the ACM Internet Measurement Conference*, pp. 393–419, 2020.

[132] P. Nurmi, S. Bhattacharya, and J. Kukkonen, *A grid-based algorithm for on-device GSM positioning*, in *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pp. 227–236, ACM, 2010.

[133] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[134] M.-R. Fida and M. K. Marina, *Impact of device diversity on crowdsourced mobile coverage maps*, in *2018 14th International Conference on Network and Service Management (CNSM)*, pp. 348–352, IEEE, 2018.

[135] U. C. Bureau, "Urban-rural classification scheme for counties." https://www.cdc.gov/nchs/data_access/urban_rural.htm, 2013. Accessed: 2021-03-01.

[136] U. C. Bureau, "Decennial census." https://www.census.gov/programs-surveys/decennial-census/decade.2010.html, 2010. Accessed: 2021-03-01.

[137] GeoPandas, "Python library for geospatial operations." http://geopandas.org/, 2019. Accessed: 2021-03-01.

[138] U.S. Census Bureau, "Definition of urban and rural." https://www.census.gov/programs-surveys/geography/guidance/geo-areas/urban-rural.html, 2019. Accessed: 2021-03-01.

[139] Z. Li, A. Nika, X. Zhang, Y. Zhu, Y. Yao, B. Y. Zhao, and H. Zheng, *Identifying value in crowdsourced wireless signal measurements*, in *Proceedings of the 26th International Conference on World Wide Web*, pp. 607–616, 2017.

[140] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *TCP Extensions for Multipath Operation with Multiple Addresses*, *RFC 6824* (2013).

[141] J. Sommers and P. Barford, *Cell vs. Wi-Fi: On the Performance of Metro Area Mobile Connections*, in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, pp. 301–314, ACM, 2012.

[142] B. Han, F. Qian, and L. Ji, *When Should We Surf the Mobile Web using both Wi-Fi and Cellular?*, in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, pp. 7–12, ACM, 2016.

[143] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, *Experimental Evaluation of Multipath TCP Schedulers*, in *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pp. 27–32, ACM, 2014.

[144] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley, *Opportunistic Mobility with Multipath TCP*, in *Proceedings of the 6th International Workshop on MobiArch*, pp. 7–12, ACM, 2011.

[145] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, *Exploring Mobile/Wi-Fi Handover with Multipath TCP*, in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular Networks: Operations, Challenges, and Future Design*, pp. 31–36, ACM, 2012.

[146] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, *Wi-Fi, LTE, or Both?: Measuring Multi-homed Wireless Internet Performance*, in *Proceedings of the 2014 Conference on Internet Measurement Conference*, pp. 181–194, ACM, 2014.

[147] B. Arzani, A. Gurney, S. Cheng, R. Guerin, and B. T. Loo, *Impact of Path Characteristics and Scheduling Policies on MPTCP Performance*, in *28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2014*, pp. 743–748, IEEE, 2014.

[148] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec, *MPTCP is Not Pareto-Optimal: Performance Issues and a Possible Solution*, in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, pp. 1–12, ACM, 2012.

[149] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, *Design, Implementation and Evaluation of Congestion Control for Multipath TCP.*, in *NSDI*, vol. 11, pp. 8–8, 2011.

[150] C. Raiciu, D. Wischik, and M. Handley, *Practical Congestion Control for Multipath Transport Protocols*, *University College London, London/United Kingdom, Tech. Rep.* (2009).

[151] K. Noda and Y. Ito, *Proposal of Novel MPTCP Congestion Control to Suppress QoS Fluctuation for WebQoE Improvement*, in *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pp. 1–3, IEEE, 2018.

[152] F. Yang, P. Amer, and N. Ekiz, *A Scheduler for Multipath TCP*, in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, IEEE, 2013.

[153] F. Yang, Q. Wang, and P. D. Amer, *Out-of-Order Transmission for In-Order Arrival Scheduling for Multipath TCP*, in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 749–752, IEEE, 2014.

[154] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, *DAPS: Intelligent delay-aware packet scheduling for multipath transport*, in *2014 IEEE International Conference on Communications (ICC)*, pp. 1222–1227, IEEE, 2014.

[155] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, *A Measurement-based Study of Multipath TCP Performance over Wireless Networks*, in *Proceedings of the 2013 Conference on Internet Measurement Conference*, pp. 455–468, ACM, 2013.

[156] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, *Detecting HTTP-based Application Layer DoS Attacks on Web Servers in the Presence of Sampling*, *Computer Networks* **121** (2017) 25–36.

[157] C.-A. Staicu and M. Pradel, *Freezing the Web: A Study of Redos Vulnerabilities in Javascript-based Web Servers*, in *27th USENIX Security Symposium USENIX Security 18)*, pp. 361–376, 2018.

[158] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "iPerf." `https://iperf.fr/`.

[159] V. Jacobson, C. Leres, and S. McCanne, "Tcpdump." `https://www.tcpdump.org/manpages/tcpdump.1.html`.

[160] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, *How Hard Can it be? Designing and Implementing a Deployable Multipath TCP*, in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, pp. 29–29, USENIX Association, 2012.

[161] P. Jurkiewicz, G. Rzym, and P. Borylo, *Flow Length and Size Distributions in Campus Internet Traffic*, *CoRR* **abs/1809.03486** (2018) [arXiv:1809.0348].

[162] FCC, "The Mobile Internet Phenomena Report 2019." `https://www.sandvine.com/hubfs/downloads/phenomena/2019-mobile-phenomena-report.pdf`, September, 2019.

[163] National Telecommunications and Information Administration, "Broadband Equity, Access, and Deployment (BEAD) Program." `https://broadbandusa.ntia.doc.gov/broadband-equity-access-and-deployment-bead-program`, May, 2022.

[164] Bharat Broadband Nigam Limited, "Planning for BharatNet Phase 2." https://bbnl.nic.in/WriteReadData/LINKS/Rpt_Nw_Plg_tool6cad24a7-eea8-4adc-91ff-468d7119eeb0.pdf, January, 2022.

[165] European Commission, "Connectivity for a European Gigabit Society." https://digital-strategy.ec.europa.eu/en/node/133/printable/pdf, April, 2022.

[166] Chawla, N. V. and Bowyer, K. W. and Hall, L. O and Kegelmeyer, W. P., *SMOTE: Synthetic Minority Over-sampling Technique*, *Journal of Artificial Intelligence Research* (2020).

[167] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, *Focal Loss for Dense Object Detection*, *CoRR* **abs/1708.02002** (2017) [arXiv:1708.0200].

[168] GSMA, "The mobile economy report." https://www.gsma.com/mobileeconomy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf, 2022.

[169] P. Schmitt and E. Belding, *Low on air: Inherent wireless channel capacity limitations*, in *ACM Workshop on Computing Within Limits*, 2017.

[170] N. V. Register, "Earthquake calls flooded 911 dispatch center." http://napavalleyregister.com/news/local/earthquake-calls-flooded-dispatch-center/article_13f12614-2589-5013-b456-c48aff5663a2.html/, September, 2014.

[171] N. Ungerleider, "Why Your Phone Doesn't Work During Disasters - and How to Fix it." https://www.fastcompany.com/3008458/why-your-phone-doesnt-work-during-disasters-and-how-fix-it/, April, 2013.

[172] G. T. 36.802, *Evolved Universal Terrestrial Radio Access (E-UTRA); NB-IOT; Technical Report for BS and UE radio transmission and reception*, February, 2016.

[173] T. Engel, "Xgoldmon." https://github.com/2b-as/xgoldmon, 2013.

[174] B. Hong, S. Park, H. Kim, D. Kim, H. Hong, H. Choi, J.-P. Seifert, S.-J. Lee, and Y. Kim, *Peeking over the Cellular Walled Gardens-A Method for Closed Network Diagnosis*, *IEEE Transactions on Mobile Computing* **17** (2018), no. 10 2366–2380.

[175] Qualcomm, "QXDM." https://www.qualcomm.com/documents/qxdm-professional-qualcomm-extensible-diagnostic-monitor, 2012.

[176] D. Spaar, "Tracing LTE on the Phone." http://www.mirider.com/weblog/2013/08/index.html, 2014.

[177] SRLabs, "SnoopSNitch." https://opensource.srlabs.de/projects/snoopsnitch, 2014.

[178] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, *MobileInsight: Extracting and Analyzing Cellular Network Information on Smartphones*, in *ACM MobiCom*, 2016.

[179] P. Schmitt, D. Iland, and E. Belding, *SmartCell: Small-scale Mobile Congestion Awareness*, *IEEE Communications Magazine* **54** (2016), no. 7 44–50.

[180] M. Tavana, A. Rahmati, and V. Shah-Mansouri, *Congestion Control with Adaptive Access Class Barring for LTE M2M Overload using Kalman Filters*, *Computer Networks* **141** (2018) 222–233.

[181] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. Wong, *D-ACB: Adaptive Congestion Control Algorithm for Bursty M2M Traffic in LTE Networks*, *IEEE Transactions on Vehicular Technology* **65** (2016), no. 12 9847–9861.

[182] A. K. Paul, H. Kawakami, A. Tachibana, and T. Hasegawa, *An AQM based Congestion Control for eNB RLC in 4G/LTE Network*, in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2016.

[183] R. Kwan, R. Arnott, R. Trivisonno, and M. Kubota, *On Pre-emption and Congestion Control for LTE Systems*, in *IEEE 72nd Vehicular Technology Conference-Fall*, 2010.

[184] P. Torres, P. Marques, H. Marques, R. Dionísio, T. Alves, L. Pereira, and J. Ribeiro, *Data Analytics for Forecasting Cell Congestion on LTE Networks*, in *2017 Network Traffic Measurement and Analysis Conference (TMA)*, IEEE, 2017.

[185] A. Chakraborty, V. Navda, V. N. Padmanabhan, and R. Ramjee, *Coordinating Cellular Background Transfers using LoadSense*, in *Proceedings of the 19th annual international conference on Mobile computing & networking*, pp. 63–74, ACM, 2013.

[186] X. Xie, X. Zhang, S. Kumar, and L. E. Li, *pistream: Physical Layer Informed Adaptive Video Streaming over LTE*, in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 413–425, ACM, 2015.

[187] X. Xie, X. Zhang, and S. Zhu, *Accelerating Mobile Web Loading using Cellular Link Information*, in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 427–439, ACM, 2017.

[188] G. T. 36.508, *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Packet Core (EPC); Common test environments for User Equipment (UE) conformance testing*, April, 2017.

[189] G. T. 36.331, *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*, January, 2016.

[190] G. T. 25.331, *Radio Resource Control (RRC); Protocol specification*, October, 2014.

[191] G. T. 24.301, *Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)*, June, 2011.

[192] G. T. 36.211, *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation*, June, 2016.

[193] G. T. 36.212, *Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding*, April, 2017.

[194] G. T. 36.304, *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode*, January, 2012.

[195] G. T. 36.300, *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2*, January, 201.

[196] S.-H. Oh and Y.-H. Kim, *Policy-based Congestion Control in WCDMA Wireless Access Networks for end-to-end QoS*, in *COIN-NGNCON 2006-The Joint International Conference on Optical Internet and Next Generation Network*, pp. 153–155, IEEE, 2006.

[197] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G evolution: HSPA and LTE for mobile broadband*. Academic press, 2010.

[198] H. Holma and A. Toskala, *WCDMA for UMTS: HSPA evolution and LTE*. John Wiley & sons, 2007.

[199] E. Research, "USRP B210." `http://www.ettus.com/all-products/UB210-KIT/`, 2010.

[200] MP Antenna, "Super-m ultra mobile antenna (25mhz–6ghz)." `https://www.mpantenna.com/product/mobile-ham-radio-antenna-scanner/`, 2014.

[201] "Text2pcap - Generate a capture file from an ASCII hexdump of packets." `https://www.wireshark.org/docs/man-pages/text2pcap.html`, 2014.

[202] G. Combs, "Tshark — Dump and Analyze Network Traffic." `https://www.wireshark.org/docs/man-pages/tshark.html`, 2012.

[203] G. T. 23.401, *General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access* , September, 2014.

[204] J. E. V. Bautista, S. Sawhney, M. Shukair, I. Singh, V. K. Govindaraju, and S. Sarkar, *Performance of CS Fallback from LTE to UMTS*, *IEEE Communications Magazine* **51** (2013), no. 9 136–143.

[205] G. T. 23.272, *Circuit Switched (CS) fallback in Evolved Packet System (EPS); Stage 2 (Release 10)*, March, 2012.

[206] 3GPP TS 38.331, "5G NR; Radio Resource Control (RRC); Protocol specification." `https://www.etsi.org/deliver/etsi_ts/138300_138399/138331/15.03.00_60/ts_138331v150300p.pdf`, 2018.

[207] S. T. Peddinti, I. Bilogrevic, N. Taft, M. Pelikan, U. Erlingsson, P. Anthonysamy, and G. Hogben, *Reducing Permission Requests in Mobile Apps*, in *Proceedings of ACM Internet Measurement Conference (IMC 19)*, 2019.

[208] T. Mangla, E. Halepovic, M. Ammar, and E. Zegura, *eMIMIC: Estimating HTTP-based Video QoE Metrics from Encrypted Network Traffic*, in *Network Traffic Measurement and Analysis Conference (TMA) 2018*, pp. 1–8.

[209] C. Gutterman, K. Guo, S. Arora, X. Wang, L. Wu, E. Katz-Bassett, and G. Zussman, *Requet: Real-time QoE Detection for Encrypted YouTube Traffic*, in *Proceedings of the 10th ACM Multimedia Systems Conference 2019*, MMSys '19.

[210] Ookla, "Mobile Speedtest Intelligence Data." `https://www.speedtest.net/reports/united-states/`, 2019.

[211] Tom's Guide, "Fastest Wireless Network 2020: It's Not Even Close." `https://www.tomsguide.com/us/best-mobile-network,review-2942.html`, 2020.

[212] N. A. Johansson, Y.-P. E. Wang, E. Eriksson, and M. Hessler, *Radio Access for Ultra-Reliable and Low-Latency 5G Communications*, in *2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 1184–1189, IEEE, 2015.

[213] The Verge, "ATT Wireless on Fake 5G Complaints." `https://www.theverge.com/2019/1/9/18175566/att-communications-ceo-fake-5g-e-complaints-response-ces-2019`, 2019.

[214] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2018-2023 White Paper." `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html`, 2018.

[215] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, *Quantification of YouTube QoE via Crowdsourcing*, in *2011 IEEE International Symposium on Multimedia*, pp. 494–499.

[216] T. Flach, N. Dukkipati, A. Terzis, B. Raghavan, N. Cardwell, Y. Cheng, A. Jain, S. Hao, E. Katz-Bassett, and R. Govindan, *Reducing Web Latency: The Virtue of Gentle Aggression*, in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, p. 159–170, 2013.

[217] J. Shaikh, M. Fiedler, and D. Collange, *Quality of Experience from User and Network Perspectives*, *Annals of telecommunications-annales des telecommunications* **65** (2010), no. 1-2 47–57.

[218] Census Bureau, "Differences Between Urban and Rural Populations." `https://www.census.gov/newsroom/press-releases/2016/cb16-210.html`, 2016.

[219] Nielsen Insights, "Streaming Consumption Rises In U.S. Markets With Early Stay-at-home Orders During COVID-19." `https://www.nielsen.com/us/en/insights/article/2020/streaming-consumption-rises-in-u-s-markets-with-early-stay-at-home-orders-during-covid-19/`, 2020.

[220] C. Midoglu, M. Moulay, V. Mancuso, O. Alay, A. Lutu, and C. Griwodz, *Open Video Datasets over Operational Mobile Networks with MONROE*, in *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, p. 426–431, 2018.

[221] V. Adarsh, "Dataset for QoE Analysis." `https://github.com/videostream-ML/urban_rural_qoe`, 2021.

[222] I. Sousa, M. P. Queluz, and A. Rodrigues, *A Survey on QoE-oriented Wireless Resources Scheduling*, *Journal of Network and Computer Applications* (2020) 102594.

[223] YouTube, "Looney Tunes Summer Vacation! WB Kids." `https://www.youtube.com/watch?v=8fKNkiJl_Ro`, 2018.

[224] 3GPP TR 29.281, *LTE General Packet Radio System (GPRS) and Tunnelling Protocol User Plane (GTPv1-U)*, July, 2018.

[225] X. Chen, N. Ding, A. Jindal, Y. C. Hu, M. Gupta, and R. Vannithamby, *Smartphone Energy Drain in the Wild: Analysis and Implications*, in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '15.

[226] Y. Freund, R. Schapire, and N. Abe, *A Short Introduction to Boosting*, *Japanese Society For Artificial Intelligence* **14** (1999), no. 771-780.

[227] L. Breiman, *Bagging Predictors*, *Machine Learning* **24** (1996), no. 2 123–140.

[228] L. Breiman, *Random Forests*, *Machine Learning* **45** (2001), no. 1 5–32.

[229] G. E. Box and D. A. Pierce, *Distribution of Residual Autocorrelations In Autoregressive-integrated Moving Average Time Series Models*, *Journal of the American statistical Association* **65** (1970), no. 332 1509–1526.

[230] Y. Freund and R. E. Schapire, *A Desicion-theoretic Generalization Of On-line Learning And An Application To Boosting*, in *European conference on computational learning theory*, pp. 23–37, Springer, 1995.

[231] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet Classification With Deep Convolutional Neural Networks*, in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[232] M. I. Jordan, *Attractor Dynamics And Parallelism In A Connectionist Sequential Machine*, in *Artificial Neural Networks: Concept Learning*, pp. 112–127. 1990.

[233] S. Hochreiter and J. Schmidhuber, *Long Short-term Memory*, *Neural computation* **9** (1997), no. 8 1735–1780.

[234] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning Phrase Representations Using RNN Encoder-decoder For Statistical Machine Translation*, *arXiv preprint arXiv:1406.1078* (2014).

[235] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine Learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825–2830.

[236] "Keras." `https://github.com/keras-team/keras`, 2019.

[237] "Tuning the Hyper-Parameters of an Estimator." `https://scikit-learn.org/stable/modules/grid_search.html`, 2019.

[238] D. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, *International Conference on Learning Representations* (2014).

[239] M. Bärtl, *YouTube Channels, Uploads and Views: A Statistical Analysis of the Past 10 Years*, *Convergence* **24** (2018), no. 1 16–32.

[240] Y. Chen, K. Wu, and Q. Zhang, *From QoS to QoE: A Tutorial on Video Quality Assessment*, *IEEE Communications Surveys & Tutorials* **17** (2014), no. 2 1126–1165.

[241] H. J. Kim and S. G. Choi, *A study on a QoS/QoE correlation model for QoE evaluation on IPTV service*, in *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, vol. 2, pp. 1377–1382, IEEE, 2010.

[242] N. Goran and M. Hadžialić, *Mathematical Bottom-to-up Approach in Video Quality Estimation based on PHY and MAC Parameters*, *IEEE Access* **5** (2017) 25657–25670.

[243] H. Chen, X. Yu, and L. Xie, *End-to-end Quality Adaptation Scheme based on QoE Prediction for Video Streaming Service in LTE Networks*, in *2013 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pp. 627–633, IEEE, 2013.

[244] S. Abdellah, M. Sara, M. N. El-Houda, and T. Samir, *QoS and QoE for Mobile Video Service over 4G LTE Network*, in *IEEE Computing Conference*, pp. 1263–1269, IEEE, 2017.

[245] T. Begluk, J. B. Husić, and S. Baraković, *Machine learning-based QoE Prediction for Video Streaming over LTE Network*, in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1–5, IEEE, 2018.

[246] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, *A Machine Learning Approach to Classifying YouTube QoE based on Encrypted Network Traffic*, *Multimedia Tools and Applications* **76** (2017), no. 21 22267–22301.

[247] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, *Measuring Video QoE from Encrypted Traffic*, in *Proceedings of the Internet Measurement Conference*, IMC, 2016.

[248] T. Mangla, E. Halepovic, M. Ammar, and E. Zegura, *MIMIC: Using Passive Network Measurements to Estimate HTTP-based Adaptive Video QoE Metrics*, in *2017 Network Traffic Measurement and Analysis Conference (TMA)*, 2017.

[249] "MONROE Video Dataset." `https://doi.org/10.5281/zenodo.1230448`, 2018.

[250] P. Casas, A. D'Alconzo, F. Wamser, M. Seufert, B. Gardlo, A. Schwind, P. Tran-Gia, and R. Schatz, *Predicting QoE in Cellular Networks using Machine Learning and in-Smartphone Measurements*, in *9th International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, 2017.

[251] D. Minovski, C. Åhlund, K. Mitra, and P. Johansson, *Analysis and Estimation of Video QoE in Wireless Cellular Networks using Machine Learning*, in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, 2019.

[252] P. Anchuen and P. Uthansakul, *Investigation into User-Centric QoE and Network-Centric Parameters for YouTube Service on Mobile Networks*, in *Proceedings of the 7th International Conference on Communications and Broadband Networking*, pp. 28–32, 2019.

[253] U. Paul, A. Ermakov, M. Nekrasov, V. Adarsh, and E. Belding, *#Outage: Detecting Power and Communication Outages from Social Networks*, in *Proceedings of The Web Conference*, WWW, p. 1819–1829, 2020.

[254] Valuates Reports, "Edge Computing Market Size, Share, Growth Trends Analysis Report By Type, By Application, and Segment Forecasts, 2021 - 2026." https://reports.valuates.com/market-reports/360I-Auto-3U70/edge-computing-market/. [Online; accessed 2021].

[255] L. Baresi and D. Filgueira Mendonça, *Towards a serverless platform for edge computing*, in *2019 IEEE International Conference on Fog Computing (ICFC)*, 2019.

[256] T. Rausch, W. Hummer, V. Muthusamy, A. Rashed, and S. Dustdar, *Towards a serverless platform for edge AI*, in *2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge)*, 2019.

[257] Z. Zhong, N. Hua, M. Tornatore, J. Li, Y. Li, X. Zheng, and B. Mukherjee, *Provisioning Short-term Traffic Fluctuations in Elastic Optical Networks*, *IEEE/ACM Transactions on Networking* (2019).

[258] Z. Liu, X. Wang, W. Pan, B. Yang, X. Hu, and J. Li, *Towards efficient load distribution in big data cloud*, in *International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2015.

[259] "Mellanox Bluefield." https://www.mellanox.com/products/BlueField-SmartNIC-Ethernet/. [Online; accessed 2021].

[260] "Broadcom Stingray." https://www.broadcom.com/products/ethernet-connectivity/network-adapters/. [Online; accessed 2021].

[261] M. Choi, S.and Shahbaz, B. Prabhakar, and M. Rosenblum, *λ-NIC: Interactive Serverless Compute on Programmable SmartNICs*, *arXiv preprint arXiv:1909.11958* (2019).

[262] M. Liu, T. Cui, H. Schuh, A. Krishnamurthy, S. Peter, and K. Gupta, *iPipe: A Framework for Building Distributed Applications on Multicore SoC SmartNICs*, in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2019.

[263] M. Liu et al., *E3: Energy-efficient Microservices on SmartNIC-accelerated Servers*, in *USENIX ATC*.

[264] D. Firestone et al., *Azure Accelerated Networking: SmartNICs in the Public Cloud*, in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2018.

[265] "Netronome Agilio CX." https://www.netronome.com/products/agilio-cx/. [Online; accessed 2021].

[266] W. L. Zwolenski M, *The digital Universe: Rich data and the increasing value of the Internet of Things*, *Journal of Telecommunications and the Digital Economy* (2014).

[267] K. Katsalis, T. G. Papaioannou, N. Nikaein, and L. Tassiulas, *SLA-driven VM Scheduling in Mobile Edge Computing*, in *IEEE 9th International Conference on Cloud Computing (CLOUD)*, 2016.

[268] P. Patel, A. H. Ranabahu, and A. P. Sheth, *Service Level Agreement in Cloud Computing*, 2009.

[269] "SmartNICs, the Next Wave in Server Acceleration.." `https://technologyevangelist.co/2020/04/04/smartnics-the-next-wave-in-server-acceleration/`. [Online; accessed 2020].

[270] L. Wang, M. Li, Y. Zhang, T. Ristenpart, and M. Swift, *Peeking behind the Curtains of Serverless Platforms*, in *USENIX Annual Technical Conference (ATC)*, 2018.

[271] I. E. Akkus, R. Chen, I. Rimac, M. Stein, K. Satzke, A. Beck, P. Aditya, and V. Hilt, *SAND: Towards High-Performance Serverless Computing*, in *2018 USENIX Annual Technical Conference*, pp. 923–935, 2018.

[272] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, *Serverless Computation with OpenLambda*, in *HotCloud*, 2016.

[273] W. Lloyd, S. Ramesh, S. Chinthalapati, L. Ly, and S. Pallickara, *Serverless computing: An investigation of factors influencing microservice performance*, in *International Conference on Cloud Engineering (IC2E)*, IEEE, 2018.

[274] "(2018) NYC taxi and limousine commission.." `https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page`. [Online; accessed 2018].

[275] A. Bhattacharjee, A. D. Chhokra, Z. Kang, H. Sun, A. Gokhale, and G. Karsai, *Barista: Efficient and scalable serverless serving system for deep learning prediction services*, *arXiv preprint arXiv:1904.01576* (2019).

[276] "Hey, HTTP(S) load generator, ApacheBench (ab) replacement, written in Go.." `https://github.com/rakyll/hey`. [Online; accessed 2019].

[277] "Stress - Tool to Impose Load on and Stress Test Systems.." `https://linux.die.net/man/1/stress`. [Online; accessed 2021].

[278] "Microsoft Azure's Public Dataset." `https://github.com/Azure/AzurePublicDataset/`, 2021.

[279] Y. Qiu, Q. Kang, M. Liu, and A. Chen, *Clara: Performance Clarity for SmartNIC Offloading*, in *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, 2020.

[280] J. Kicinski and N. Viljoen, *eBPF Hardware Offload to SmartNICs: cls bpf and XDP*, *Proceedings of netdev* (2016).

[281] A. Mehta, J. Dürango, J. Tordsson, and E. Elmroth, *Online Spike Detection in Cloud Workloads*, in *IEEE international conference on cloud engineering*, 2015.

[282] V. Nikolov, S. Kächele, F. J. Hauck, and D. Rautenbach, *Cloudfarm: An Elastic Cloud Platform with Flexible and Adaptive Resource Management*, in *IEEE/ACM 7th International Conference on Utility and Cloud Computing*, 2014.

[283] A. Gandhi, T. Zhu, M. Harchol-Balter, and M. A. Kozuch, *SOFTScale: Stealing Opportunistically for Transient Scaling*, in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, Springer, 2012.