Performance-aware Cellular Networking through Measurement-Driven, Cross-Layer Control

by

Ahmad Hassan

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER ENGINEERING)

May 2026

Dedication

To my family.

Acknowledgements

A Ph.D. is often described as a marathon, but mine felt less like a steady race and more like a long, winding journey — full of detours, steep climbs, and moments where the view made everything worth it. Over the past five years, this journey has taken me across two universities, two cities, and countless experiences that have shaped not only how I approach research, but how I think, work, and live. There were highs that made the effort feel effortless, and lows that tested patience and resilience. Looking back, it is these ups and downs that made this experience meaningful and transformed me into the person I am today.

I would like to begin by thanking my advisor, Feng Qian, who has been much more than just an academic mentor — an academic parent in every sense. His ability to understand what a project needs at any given time has always amazed me. Even more importantly, he understood what *I* needed at different stages of my journey: close guidance and structure in the early years, and complete trust and freedom in the later stages. His work ethic, clarity of thought, and commitment to high-quality research have been a constant source of inspiration.

I am also deeply grateful to my thesis committee members, Prof. Bhaskar Krishnamachari and Prof. Mengyuan Li. My interactions with Bhaskar, though limited, have always been thoughtful and encouraging, and I truly appreciate his support over the years. Mengyuan introduced me to

new research directions, particularly in the networking aspects of confidential computing, and I am thankful for the perspective and curiosity he brought into my work.

I have been fortunate to collaborate closely with outstanding researchers throughout my Ph.D. I would like to thank Prof. Zhi-Li Zhang and Prof. Z. Morley Mao for their invaluable guidance and insights. Zhi-Li's passion for research and his openness to exploring new ideas have always inspired me. Morley, my academic grandparent, is truly a rockstar, and working with her has been both inspiring and humbling.

I have also had the opportunity to work with some amazing collaborators: Wei Ye, Anlan Zhang, Rostand A. K. Fezeu, Jason Carpenter, Ruiyang Zhu, Shuowei Jin, Arvind Narayanan, Xumiao Zhang, Eman Ramadan, and Bin Hu. I am equally grateful to my labmates across UMN and USC — Faaiq Bilal, Zejun Zhang, Ritvik Janamsetty, and Yu Liu — for the discussions, camaraderie, and shared experiences along the way.

My internships gave me a completely different perspective on research and collaboration. At HPE Labs, Puneet Sharma has been a manager, mentor, and a friend. I will always cherish the Friday evening soccer, volleyball, and cricket games—along with all the friendly banter haha. Shivang Aggarwal and Mohamed Ibrahim were incredible mentors who helped shape some of the most important parts of my research. At Samsung Research America, I had the opportunity to work under Vutha Va, whose dedication and tenacity left a strong impression on me. That experience also taught me how to navigate challenging situations with patience and effective communication. I am also thankful to Anum Ali and Ethan for being great collaborators — I have always valued Anum's perspective and advice.

Beyond research, I always tried to maintain a balance between work and life, and I have been incredibly lucky to build meaningful friendships along the way. In Minneapolis, I am grateful for

Ali, Faizal, Ahmed, Bakhty, Salman, Lalit, Rushikesh, Nanditha, and Dustin. In Los Angeles, my roommates Patrick, Philip, and David made the last two years truly special — and of course our cat, Kevin, the most important member of our house. LA is also where I met my girlfriend Cecilia and Jazz, her dog, and I am excited for everything that lies ahead with them. I am thankful for my Pakistani friends Zoha, Shanza, and Namra, as well as my VGSA friends Natalie, Jorge, Michael, Malia, and Brian. My pickleball, cricket, badminton and gym crews made sure I stayed active and sane. If I have missed anyone, please know that I am equally grateful to you.

My time at UMN and USC gave me two very different, yet equally valuable experiences. Minneapolis had the harshest winters, but also a focused and supportive research environment — a perfect place to grow as a researcher. Los Angeles, on the other hand, brought sunshine, energy, and a completely different pace of life. It came with distractions, but also with experiences I would never trade for anything.

Finally, and most importantly, I want to thank my family. None of this would have been possible without their unwavering support, love, and sacrifices. My parents, especially my mother, have been the foundation of everything I have achieved. She is the hardest working person I know, and my struggles during the Ph.D. are insignificant compared to what she has done for our family. My father's dedication and support toward my education continue to inspire me to strive for excellence and push myself further in everything I do. I can never thank my parents enough. My sister has always been the first person I turn to when I want to share something — I admire her intelligence, discipline, and work ethic. Missing her wedding was one of the hardest sacrifices during my Ph.D. journey. My brother has been my constant source of comfort — talking to him always brought me back to balance when things felt overwhelming.

This dissertation is dedicated to you — my family.

Table of Contents

Dedication ii

Acknowledgements iii

List of Tables x

List of Figures xii

Abstract xvii

Chapter 1: Introduction 1
- 1.1 *Mobility Matters Most*: Measurement, Modeling, and Mitigation of 5G Handovers 4
- 1.2 *Beyond Static Resource Adaptation*: Dynamic Duplexing Under Asymmetric 5G Workloads 5
- 1.3 *Exploiting Multi-Cell Diversity*: Performance-Driven Connectivity Management in 5G 7
- 1.4 *From Periodic Scanning to Predictive Control*: Energy-Aware Idle Measurement Adaptation 9
- 1.5 Thesis Organization 11

Chapter 2: Cellular Resource Management Foundations 13
- 2.1 Cellular Architecture Overview 14
- 2.2 Mobility Management in 4G and 5G Networks 19
- 2.3 Duplex Resource Allocation 25
- 2.4 Multi-Cell Connectivity and CM Procedures 29
- 2.5 Idle-State RRM Measurements 32
- 2.6 Structural Characteristics of Current Mechanisms 36

Chapter 3: Measurement, Modeling, and Mitigation of 5G Handovers 39
- 3.1 Introduction 39
- 3.2 Measurement Methodology 44
- 3.3 Impact of Mobility on Application Performance 48
  - 3.3.1 Quantifying Application QoE under Mobility 48
  - 3.3.2 *5G-only vs. dual* traffic mode in NSA Deployments 51

3.4 Characteristics of 5G Handovers . . . 52
3.4.1 Handover Frequency . . . 52
3.4.2 Handover Duration . . . 53
3.4.3 Handover Energy Consumption . . . 55
3.5 Implications of 5G Handovers on Carriers . . . 56
3.5.1 Coverage Landscape in 5G . . . 57
3.5.2 Impact of 5G HOs on Bandwidth . . . 58
3.5.3 Impact of eNB and gNB Co-location . . . 59
3.6 4G/5G Handover Prediction . . . 60
3.6.1 Challenges and Goals . . . 60
3.6.2 System Design . . . 61
3.6.3 Performance Evaluation . . . 64
3.6.4 Prognos Use Cases . . . 66
3.7 Summary and Broader Implications . . . 67

Chapter 4: Dynamic Duplexing Under Asymmetric 5G Workloads . . . 70
4.1 Introduction . . . 70
4.2 Motivation & Challenges . . . 72
4.2.1 Experiment Setup . . . 73
4.2.2 Need for Dynamic TDD Policy Adjustment . . . 74
4.2.3 Challenges . . . 77
4.3 Design Overview . . . 78
4.4 Proactive Demand Customization . . . 80
4.4.1 Cross-layer BS-level Feature Engineering . . . 80
4.4.2 Context-aware Resource Forecasting . . . 82
4.4.3 RL Agent Architecture and Training . . . 83
4.5 Context-aware Policy Provision . . . 84
4.5.1 Conservative Policy Smoothing . . . 84
4.5.2 QoS-aware TDD Policy Derivation . . . 85
4.6 Wixor Implementation . . . 88
4.7 Evaluation . . . 90
4.7.1 Experiment Setup . . . 90
4.7.2 Overall Benefit for the Applications . . . 92
4.7.3 Over-the-air Evaluation of Wixor . . . 94
4.7.4 Wixor under Diverse Settings . . . 96
4.7.5 Wixor Deep Dive . . . 98
4.7.6 Micro-benchmarking . . . 101
4.8 Discussion & Conclusion . . . 102

Chapter 5: Performance-Driven Connectivity Management in 5G . . . 106
5.1 Introduction . . . 106
5.2 Measurement and Motivation . . . 111
5.2.1 Measurement Setup . . . 111
5.2.2 Wide Availability of Cell Deployments . . . 112
5.2.3 Performance Diversity in Cell Deployments . . . 113

5.3 Opportunistic Performance-driven Connectivity Management (OPCM) . . . 116
5.3.1 Design Overview . . . 117
5.4 Smart Decision Framework . . . 119
5.4.1 Cell Set Generation . . . 119
5.4.2 RAN Policy Compliance via Cell Set Pruning . . . 121
5.4.3 Opportunistic Decision Making . . . 122
5.5 Hybrid Profiling Engine . . . 124
5.5.1 Passive Performance Approximation . . . 124
5.5.2 OPCM Performance Criteria . . . 126
5.6 Robust Execution Module . . . 128
5.7 Implementation . . . 129
5.8 Evaluation . . . 131
5.8.1 Experimental Setup . . . 132
5.8.2 OPCM QoE Improvement . . . 134
5.8.3 OPCM Benchmarking . . . 137
5.8.4 End-to-end System Evaluation . . . 138
5.8.5 Micro-benchmarks . . . 141
5.9 Discussion & Conclusion . . . 144

Chapter 6: Energy-Aware Idle Measurement Adaptation . . . 147
6.1 Introduction . . . 147
6.1.1 Motivating Prediction-Aware Measurement Scheduling . . . 149
6.1.2 Limitations of Existing Measurement Relaxation . . . 150
6.1.2.1 Predictable Mobility with Short-Term Fluctuations . . . 151
6.1.2.2 Stable Serving Cell with Higher-Priority Neighbors . . . 152
6.1.3 Towards Adaptive Measurement Relaxation . . . 153
6.2 Proposed Solution . . . 154
6.2.1 Measurement Prediction and Relaxation Cost . . . 156
6.2.2 Energy Saving Estimation . . . 163
6.2.2.1 Empirical Power Profiling . . . 164
6.2.2.2 Configuration-Based Estimation . . . 165
6.2.3 Utility Computation . . . 166
6.2.4 Adaptive Relaxation Mechanism . . . 167
6.2.4.1 Progressive Adaptation Strategy . . . 168
6.2.4.2 Full Search Strategy . . . 169
6.2.5 Standby Evaluation Module . . . 169
6.3 Evaluation . . . 170
6.3.1 Experiment Setup . . . 171
6.3.2 Dissecting Energy Saving Gains . . . 172
6.3.3 End-to-end Results . . . 175
6.3.4 Micro-benchmarking . . . 177
6.4 Conclusion and Future Directions . . . 178

Chapter 7: Related Work . . . 180

7.1 Mobility Management and Handover Prediction 180
7.2 Duplex Resource Allocation 182
7.3 Connectivity Management 184
7.4 Idle-State Measurement Adaptation 187

Chapter 8: Conclusion and Future Work 189
8.1 Summary of Contributions 190
8.2 Future Research Directions 191
8.3 Concluding Remarks 193

Bibliography 195

List of Tables

2.1 LTE/NR measurement reporting events used for mobility decisions. M_S and M_N denote measurements of the serving and neighboring cells, respectively. 20

2.2 Classification of handover procedures observed in LTE, NSA 5G, and SA 5G. . . . 23

2.3 An overview of common CM procedures and their UE data transmission modes (idle, inactive, connected), actions (add, modify, remove cells), and criteria (cell accessibility, link quality, absolute priority). 31

3.1 Dataset overview: network footprint, mobility events, and trace coverage. 41

3.2 Prognos performance evaluation on D1 and D2. 64

4.1 Application workloads used in this paper (UL = uplink, DL = downlink; Lat. Sen. = latency-sensitive, Bwd. Int. = bandwidth-intensive). 74

4.2 TDD policies used by major public 5G operators in the U.S. 75

4.3 5G QoS Identifier (5QI) values used for evaluated applications. 90

4.4 Overall QoE comparison across six application workloads over 6+ hours of channel traces. Values denote mean $\pm$ standard deviation. Arrows indicate improvement direction; shaded cells denote best performance. 93

4.5 System overhead at 65% traffic load. 95

4.6 NN inference time (mean $\pm$ std). 95

4.7 Average BS throughput gap (%) between Wixor and baselines. 97

4.8 Wixor performance across common 5G numerologies (μ). 97

4.9 Comparing different RL schemes. 101

4.10 Varying number of neurons and filters (1D-CNN unit). 102

4.11 Varying number of hidden layers. 102

5.1 Comparison of existing CM techniques with OPCM. 109

5.2 Cell combinations used in our experiments. The number of carriers (SCells) may vary over time. 114

5.3 OPCM RAN objective examples. 121

5.4 Comparing system overhead (30 users). 140

5.5 OPCM Profiling Engine vs. Baselines. 143

6.1 Default parameters used for evaluating adaptive measurement relaxation. 171

6.2 Best- and worst-case inter-frequency relaxation scenarios for N25 and N41 bands. 173

6.3 Prediction performance of different models across relaxation levels. Each cell reports MAE / RMSE / Correlation, with lower error and higher correlation indicating better performance. 178

List of Figures

1.1 Overview of this dissertation. 3

2.1 Simplified Radio Access Network (RAN) architecture under Non-Standalone (NSA) and Standalone (SA) deployments. 14

2.2 Simplified 5G NR protocol stack highlighting user/data plane and control plane layers. 16

2.3 RRC state machine in 5G NR. 17

2.4 Logical phases of a connected-mode handover in 5G. 22

2.5 SCG HO procedures for mobility in NSA 5G. 24

2.6 5G frame structure and TDD pattern for numerology $\mu = 1$. 26

2.7 Overview of radio resource scheduling in 5G TDD networks. 27

2.8 Example of a UE configured with a Primary Cell (PCell) and multiple Secondary Cells (SCells) under carrier aggregation. 30

2.9 Illustration of a Paging Occasion (PO) in 5G NR. The UE periodically wakes to monitor paging messages and perform configured measurements, leading to current spikes. 33

2.10 Zooming into the idle-state 5G NR paging occasion. 34

2.11 Measurement scheduling in idle mode. Top: default operation ($R = 1$) where measurements occur at every PO. Bottom: relaxed operation ($R = 2$) where measurements occur less frequently. 35

3.1 An overview of our measurement setup. 46

3.2 Video conferencing latency and packet loss during HOs in NSA 5G (Low-Band). . 49

3.3 Cloud gaming latency and frame drop rate during HOs in NSA 5G. 49

3.4 Impact of HOs and radio band on the QoE of volumteric video streaming. 50

3.5 TCP (BBR) RTT during HOs in two NSA deployment modes. 52

3.6 HO preparation stage (T_1) for T-Mobile in NSA 5G *vs.* SA 5G *vs.* LTE. 54

3.7 Comparison of HO execution stage T_2 across access technologies (NSA 5G *vs.* SA 5G *vs.* LTE) and radio bands (Low-Band *vs.* mmWave). 55

3.8 Comparing power consumption of HOs in Low-Band NSA 5G *vs.* mmWave NSA 5G *vs.* Mid-Band LTE. 56

3.9 Comparison of tower's effective coverage footprint (diameter): with and without NSA. 57

3.10 Impact of SCGC on network bandwidth in 5G mmWave. 58

3.11 Handover Duration (T_1 + T_2) with same (vs. different) 4G-LTE PCI and 5G-NR PCI. 59

3.12 Design of HO prediction system Prognos. 62

3.13 Lead-time improvement in HO prediction enabled by *report predictor*. 65

3.14 Impact of bootstrapping with most frequent pattern during startup phase of Prognos. 66

3.15 QoE improvement due to Prognos for *16K panoramic VoD* and *real-time volumetric video streaming*. 67

4.1 The high variability in traffic load observed from two public 5G networks. 75

4.2 Impact of static TDD policies on the live video ingest application's QoE (sending bitrate and ingest delay). 76

4.3 Effect of *inter-slot delay* on Edge Video Analytics (EVA) frame response latency under different background traffic settings. 76

4.4 Impact of frequent TDD policy updates on TCP congestion control behavior. . . . 77

4.5 Comparison of one-way UL and DL latencies in the testbed (log-scale). 77

4.6 A high-level overview of Wixor. 79

4.7 Context-aware resource forecasting using reinforcement learning. 83

4.8 BS QoS metrics for the simulation experiments in Table 4.4. 92

4.9 Comparison of Edge Video Analytics QoE across baselines. 94

4.10 Comparison of Video-on-Demand streaming QoE across baselines. 94

4.11 Wixor's impact on RAN metrics. 98

4.12 System scalability under multiple users. 99

4.13 Performance gap between Wixor and *Oracle*. 99

4.14 Prediction accuracy of the demand customization engine compared with *DRP*. . . 100

4.15 Evaluation of the slot derivation module. 100

4.16 Evaluating conservative policy smoothing. 101

5.1 Decoupling CM criteria and underlying CM procedures via an abstraction layer. . 107

5.2 Density of cell deployments across different regions. 113

5.3 A case study to quantify the performance gap between legacy CM and the highest performing cell combinations. The color gradient ranges from red (highest performance) to yellow (lowest performance). 115

5.4 Performance breakdown of cell combinations in our dataset. 115

5.5 The overall design workflow of OPCM. 117

5.6 Group-based addition of carriers during CA. 120

5.7 Example of correlation between two cell combinations. 125

5.8 Time-Lagged Cross-Correlation (TLCC) across cell combinations. 125

5.9 Custom metric registration flow in OPCM. 131

5.10 The over-the-air prototype testbed of OPCM. 132

5.11 DL throughput density in 60m$\times$40m test area. 132

5.12 Comparing OPCM VoD streaming performance across baselines. 135

5.13 Comparing OPCM video analytics performance across baselines. 135

5.14 Comparing OPCM video ingest performance across baselines. 136

5.15 Comparing OPCM energy efficiency across baselines. 136

5.16 Benchmarking OPCM performance. 138

5.17 Evaluating OPCM compliance with legacy link quality criterion. 138

5.18 Comparing RAN metrics across various load conditions. 139

5.19 OPCM performance under different mobility scenarios. 139

5.20 OPCM scalability as the number of users increases. 140

5.21 Decision framework vs. Oracle. 142

5.22 Comparing OPCM execution module with legacy. 142

5.23 Impact of $\boldsymbol{\lambda}$ on OPCM performance. 143

6.1 RSRP trace from a walking experiment. Green: strong, stable signal; red: degrading region. Dashed gray line shows underlying trend; dotted line marks reselection threshold (-97 dBm). 151

6.2 Stationary experiment. Serving cell (N25) remains strong; higher-priority N41 neighbors remain below reselection threshold. Continuous inter-frequency measurements occur without reselection events. 153

6.3 High-level workflow of the adaptive measurement relaxation framework, showing its five core components and their interactions. 155

6.4 Collection of historical measurements aligned to the relaxation factor to form the measurement buffer used for prediction. 157

6.5 Decomposition-based prediction module. Measurements from the buffer are split into trend and residual components, predicted separately, and then combined to produce $\hat{x}_t$. 160

6.6 Intra- and inter-frequency measurements within a PO. 163

6.7 Example power traces from Monsoon profiling showing current spikes for measurement events overlayed on top of the PO. 164

6.8 PO durations and per-PO energy savings for bands N25 and N41. 165

6.9 PO duration and energy-saving bounds for inter-frequency measurement relaxation. 173

6.10 Estimated best- and worst-case energy savings for N25 across relaxation factors R. $R = \infty$ represents the upper bound where all inter-frequency measurements are removed. 174

6.11 Estimated best- and worst-case energy savings for N41 across relaxation factors R, showing smaller gains than N25 due to shorter paging occasions and fewer inter-frequency measurements. 174

6.12 Adaptive relaxation dynamics under a low-mobility walking trace. 175

6.13 Overall utility (U_t^R) across the full trace. 176

Abstract

Performance-aware Cellular Networking through

Measurement-Driven, Cross-Layer Control

by

Ahmad Hassan

Chair: Feng Qian

Modern cellular networks are designed to support latency-sensitive, bandwidth-intensive, and energy-constrained applications, yet many control-plane mechanisms remain driven by static, signal-strength-based heuristics originally intended for coverage and reliability. Although 5G New Radio (NR) introduces substantial flexibility—through carrier aggregation, dual connectivity, flexible numerologies, beamforming, and dynamic Time Division Duplexing (TDD)—core control surfaces such as mobility management, duplex allocation, connectivity management, and idle-mode measurement scheduling largely operate reactively and independently, without explicitly optimizing system-level objectives such as application Quality of Experience (QoE), Quality of Service (QoS), fairness, or user equipment (UE) energy efficiency. As a result, there exists a growing mismatch between the capabilities of modern cellular networks and the performance demands of emerging applications. This dissertation shows that many of these limitations stem from a lack of measurement-driven and cross-layer reasoning in control-plane decisions. It demonstrates that

incorporating prediction, cross-layer signals, and explicit system-level objectives into decision-making can substantially improve performance, while remaining compatible with existing 3GPP procedures and deployment constraints. Rather than redesigning the cellular stack, this work focuses on augmenting existing mechanisms with lightweight, practical intelligence grounded in real-world measurements. We study this across four domains. First, a 6,200 km cross-layer measurement campaign reveals that 5G handovers are frequent and can significantly degrade application QoE and UE energy efficiency; we design Prognos, a predictive mobility framework that anticipates handovers and mitigates their impact. Second, we show that static or reactive TDD policies are misaligned with asymmetric workloads and introduce Wixor, a predictive, QoS-aware dynamic TDD control system that jointly adapts slot distribution and arrangement. Third, we characterize heterogeneous multi-cell deployments and uncover substantial performance diversity across feasible cell combinations; we develop OPCM, a performance-driven connectivity management framework that improves throughput, latency, and energy efficiency while respecting operator policies and ensuring scalability across users. Finally, we demonstrate that rule-based idle-mode measurement relaxation leaves significant energy savings unrealized, and propose PARMA, a prediction-based adaptive framework that balances energy efficiency against reselection risk through context-aware measurement control. Across all domains, prototypes and large-scale evaluations demonstrate measurable improvements in throughput, latency, QoE, and energy efficiency. Overall, this dissertation highlights the importance of measurement-driven, cross-layer design in improving practical cellular network performance, and shows that smarter control—rather than new radio features—is key to unlocking the full potential of modern cellular networks under real-world constraints.

Chapter 1

Introduction

Modern cellular networks support latency-sensitive, bandwidth-intensive, and mission-critical applications. The transition from 4G LTE to 5G New Radio (NR) has introduced substantial advances at the physical and link layers, including flexible numerologies, carrier aggregation (CA), dual connectivity (DC), massive MIMO, beamforming, and dynamic Time Division Duplexing (TDD) [2, 3, 1]. These enhancements significantly expand the design space for radio resource allocation and network performance optimization.

At the same time, application workloads have undergone a structural shift. Emerging use cases, such as cloud gaming, augmented and virtual reality, volumetric video streaming, edge-assisted perception, and real-time analytics, place **stringent and often asymmetric requirements** on latency, uplink capacity, throughput stability, and energy efficiency [75, 28, 70, 191]. Unlike traditional mobile traffic, these workloads are sensitive not only to average throughput but also to jitter, transient disruptions, uplink bottlenecks, and control-plane overhead.

Despite this evolution, many cellular resource management mechanisms continue to operate using **static rules and rigid heuristics** that were originally designed to ensure link reliability, coverage, and simplicity. These include mobility management (handover triggering), duplex

resource allocation, connectivity management (cell selection and carrier aggregation), and idle-mode measurement scheduling. Their logic is largely governed by link-quality indicators and fixed thresholds. While effective for maintaining stable connections, these mechanisms do not explicitly optimize for system-level objectives such as application Quality of Experience (QoE), network Quality of Service (QoS), or user equipment (UE) energy efficiency, and are typically designed in isolation with limited coordination across control surfaces.

Modern cellular networks expose increasingly rich **cross-layer signals** across the PHY, MAC, RRC, transport, and application layers. Base stations and UEs can observe channel quality indicators (CQI), buffer occupancy, measurement reports, transport dynamics, and application-level performance. However, these signals are often underutilized in decision-making. In practice, control-plane mechanisms remain largely reactive and localized, rather than leveraging this information to make proactive, system-aware decisions.

This gap becomes more pronounced under contemporary workloads. Mobility decisions based solely on signal strength can degrade application performance even when radio conditions improve. Static or downlink-biased duplexing policies can under-provision uplink resources for emerging applications. Heterogeneous deployments offer multiple viable cell combinations with diverse performance characteristics, yet connectivity decisions remain signal-driven. Similarly, idle-mode measurements can incur avoidable energy overhead in predictable conditions.

This dissertation shows that many of these inefficiencies stem from the lack of *measurement-driven and cross-layer reasoning* in control-plane decisions. Through multiple case studies, it demonstrates that incorporating prediction, cross-layer signals, and system-level objectives into decision-making can significantly improve application performance, network efficiency, and UE energy consumption, while remaining compatible with existing 3GPP procedures.

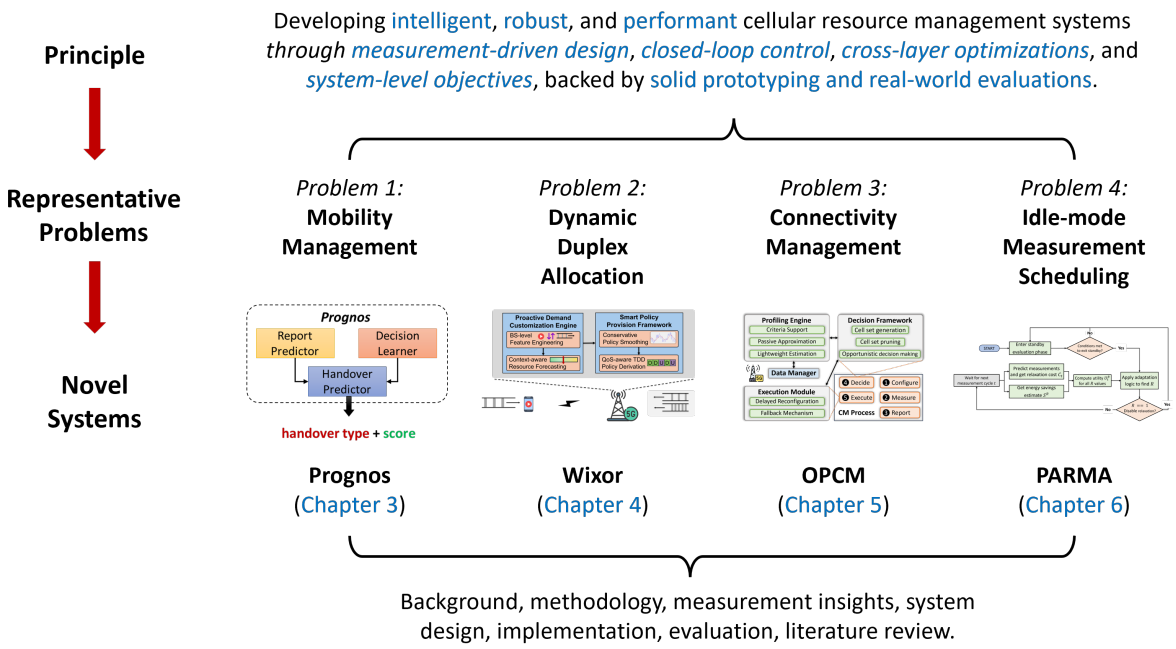


Figure 1.1: Overview of this dissertation.

Figure 1.1 summarizes the scope of this work. Across multiple control surfaces in operational 5G networks—including mobility management, duplex resource allocation, connectivity management, and idle-mode measurement scheduling—we first identify limitations of existing rule-based mechanisms through empirical analysis. We then design predictive, cross-layer systems that incorporate system-level objectives directly into the decision process. The resulting systems (Chapters 3–6) demonstrate measurable improvements in QoE, throughput, latency, and energy efficiency, without requiring changes to the cellular standard.

In the following sections, we introduce the representative problems addressed in this dissertation and summarize the corresponding system designs.

1.1 *Mobility Matters Most*: Measurement, Modeling, and Mitigation of 5G Handovers

Mobility management is one of the most fundamental control-plane functions in cellular networks. As UE moves across cells, handover procedures ensure continuity of connectivity and service. In 5G deployments, however, mobility management has become substantially more complex due to heterogeneous frequency bands, dense small-cell deployments, and the coexistence of standalone (SA) and non-standalone (NSA) architectures. These architectural changes increase both the frequency and diversity of handover events.

To understand the implications of this complexity, we conducted a large-scale cross-layer measurement study spanning more than 6,200 km of drive tests across three major U.S. carriers, collecting over 600 GB of logs and observing more than 47,000 handover events [72]. The dataset spans 4G, 5G NSA, and 5G SA deployments, covering low-band, mid-band, and mmWave frequencies. This measurement campaign provides a comprehensive view of mobility behavior in operational 5G networks.

Our analysis reveals that handovers in 5G are both more frequent and more disruptive than in previous generations. For example, 5G handovers occur approximately every 0.4 km on average, compared to 0.6 km in 4G, with mmWave deployments exhibiting even higher rates. These handovers are not benign: during live video conferencing, frame loss increases by 2.24× and latency can rise by up to 14.5×; in cloud gaming at 4K 60 FPS, dropped frames increase by 3.64× during handovers. Additionally, frequent NSA handovers introduce measurable energy overhead due to additional signaling and control-plane transitions.

Importantly, these degradations occur even when handover decisions are triggered by stronger signal conditions. In some cases, throughput decreases following handover despite improvements in radio-layer metrics. This disconnect illustrates a central theme of this dissertation: mobility decisions driven solely by signal strength and threshold-based event logic do not necessarily align with system-level performance objectives.

The logical structure of handover procedures (i.e., configuration, measurement, reporting, decision, and execution) naturally lends itself to a closed-loop control interpretation. However, in current deployments, the loop is predominantly reactive and signal-driven. To address this limitation, we develop Prognos, a predictive handover framework that incorporates cross-layer signals to anticipate mobility events before they occur. Prognos decomposes the problem into two stages: predicting future signal evolution and learning base-station decision logic, enabling proactive mitigation of application-level disruptions.

By reformulating mobility management as a measurement-driven, predictive control problem, this work demonstrates that handover-related performance degradation is not an unavoidable artifact of mobility, but a consequence of reactive decision-making. The results show that predictive mobility control can measurably improve application-level QoE while remaining compatible with existing 3GPP procedures.

1.2 *Beyond Static Resource Adaptation*: Dynamic Duplexing Under Asymmetric 5G Workloads

Time Division Duplexing (TDD) is a central mechanism for resource allocation in contemporary 5G deployments. Unlike earlier generations that predominantly relied on Frequency Division

Duplexing (FDD), a large fraction of 5G operators deploy TDD in mid-band and mmWave spectrum to enable flexible uplink (UL) and downlink (DL) sharing of the same frequency band [76]. In principle, dynamic TDD allows the base station to adjust the distribution of UL and DL time slots in response to traffic demand. In practice, however, deployed TDD configurations are often static or semi-static and heavily biased toward downlink traffic.

This design assumption reflects historical mobile broadband usage patterns, where downlink traffic dominated. However, emerging 5G workloads exhibit increasingly asymmetric and dynamic traffic characteristics. Applications such as live video ingestion, edge-assisted perception, real-time analytics, and interactive AR/VR generate substantial uplink demand and impose strict latency constraints [31, 27, 97, 24, 100]. Under such conditions, static or DL-heavy TDD policies can under-provision uplink resources, leading to queue buildup, elevated latency, and degraded application performance.

To examine this issue, we conducted a measurement study of operational 5G networks and observed that major public deployments employ nearly fixed TDD slot distributions, regardless of instantaneous traffic fluctuations. Our analysis shows that base station traffic load varies rapidly over short time scales, yet TDD slot allocations remain largely unchanged. Furthermore, even when dynamic TDD mechanisms are available, straightforward reactive strategies, such as adjusting slot ratios based solely on recent traffic demand, fail to capture short-term dynamics and may interfere with higher-layer congestion control.

Importantly, duplex resource allocation is not solely a matter of UL/DL slot percentage. The temporal arrangement of slots within a TDD pattern influences inter-slot delay and therefore directly impacts latency-sensitive applications. Additionally, frequent reconfiguration of TDD policies can destabilize transport-layer rate adaptation and introduce performance oscillations.

These observations suggest that duplexing decisions must account for cross-layer interactions spanning the PHY, MAC, transport, and application layers.

We reformulate duplex resource allocation as a measurement-driven, closed-loop control problem. We introduce Wixor, a dynamic TDD adaptation system that (i) predicts short-term UL and DL demand using base-station-level features, (ii) optimizes both slot distribution and arrangement under practical constraints, and (iii) applies conservative smoothing to preserve transport-layer stability. Unlike purely theoretical or simulation-driven analyses, Wixor is prototyped on a programmable 5G testbed and evaluated using both over-the-air experiments and trace-driven simulations. The results demonstrate that aligning duplex decisions with system-level performance objectives yields measurable improvements in throughput and latency across diverse workloads while remaining compatible with 3GPP signaling and deployment constraints.

This case study illustrates that duplexing, like mobility management, benefits from being treated not as a static configuration problem but as a predictive, cross-layer control mechanism operating under explicit performance objectives.

1.3 *Exploiting Multi-Cell Diversity*: Performance-Driven Connectivity Management in 5G

Modern 5G deployments exhibit substantial heterogeneity across frequency bands, radio access technologies, carrier aggregation configurations, and deployment architectures. A single UE may simultaneously observe multiple candidate cells across low-, mid-, and high-band frequencies, as well as across SA and NSA configurations. Operators further employ carrier aggregation and

dual connectivity to combine multiple component carriers, creating a large space of possible cell combinations [45, 189].

This architectural richness introduces a fundamental opportunity: different cell combinations can exhibit significantly different performance characteristics in terms of throughput, latency, stability, and energy efficiency. However, legacy connectivity management (CM) mechanisms, encompassing cell selection, reselection, handover, and carrier aggregation, are primarily driven by signal strength and frequency priority rules. While effective for maintaining coverage and link reliability, these mechanisms do not explicitly optimize for system-level performance objectives.

To quantify this gap, we conducted large-scale measurement campaigns across multiple cities and countries, systematically enumerating feasible cell combinations at each location. Our experiments reveal that UEs are often surrounded by multiple base stations and numerous valid cell combinations, with substantial performance diversity across them. In many cases, the cell combination selected by legacy signal-based CM does not provide the highest achievable throughput, lowest latency, or best energy efficiency. This observation suggests that connectivity management is not merely a connectivity problem, but an optimization problem over a heterogeneous resource space. Selecting a serving cell (or a combination of cells) should not be reduced to choosing the strongest signal, but instead framed as selecting the configuration that best aligns with system-level objectives under operator constraints.

However, reformulating connectivity management as an optimization problem introduces new challenges. First, the number of possible cell combinations can grow combinatorially with carrier aggregation and dual connectivity. Second, decisions must respect radio access network (RAN) policies such as fairness, load balancing, and spectrum usage constraints. Third, frequent

switching between cell combinations may introduce transient disruptions. These constraints require a coordinated and deployment-aware design.

In this chapter, we present OPCM, a centralized, performance-driven connectivity management framework that operates at the base station. OPCM constructs a pruned set of feasible cell combinations, profiles their performance using a hybrid measurement strategy, and opportunistically selects configurations that improve throughput, latency, or energy efficiency while satisfying RAN policy constraints [74]. Unlike UE-side approaches, OPCM coordinates decisions across multiple UEs and preserves fairness and operator compliance.

By exploiting multi-cell diversity through measurement-driven and cross-layer decision-making, this work demonstrates that connectivity management can be transformed from a signal-driven heuristic into a structured control problem over heterogeneous resources. This case further reinforces the dissertation's central thesis: diverse cellular resource management mechanisms can be systematically reformulated as closed-loop control systems that optimize system-level objectives.

1.4 *From Periodic Scanning to Predictive Control*: Energy-Aware Idle Measurement Adaptation

Idle-mode operation constitutes a substantial portion of a mobile device's lifetime. Even when not actively transmitting data, UEs must periodically wake up to monitor paging occasions and perform radio resource management (RRM) measurements for cell reselection. These measurements include intra-frequency and inter-frequency scans of neighboring cells, as mandated by 3GPP mobility procedures. While necessary for maintaining mobility robustness, such periodic scanning introduces non-trivial energy overhead.

In current deployments, idle-mode measurement scheduling follows rule-based relaxation mechanisms. Measurement frequency may be reduced under coarse conditions such as low mobility or strong serving-cell signal quality. However, these rules are static and conservative: they rely on fixed thresholds and predefined mobility classifications rather than fine-grained, predictive reasoning. As a result, UEs often continue performing inter-frequency measurements even in scenarios where channel conditions evolve slowly and reselection events are highly unlikely.

Through controlled experiments using commercial 5G devices instrumented with power monitors and lower-layer logs, we observe that repeated inter-frequency scans can extend paging-related wake durations and contribute measurable idle-mode energy consumption. In stable channel conditions, these scans rarely trigger reselection events, indicating a mismatch between measurement frequency and actual mobility risk.

This inefficiency highlights a deeper structural limitation: idle measurement scheduling is treated as a periodic compliance task rather than as a control problem. The UE passively executes network-configured measurement intervals, with limited ability to reason about short-term channel predictability or reselection probability. Yet modern devices possess sufficient computational capability and cross-layer context—including historical signal traces and mobility cues—to enable predictive decision-making.

In this chapter, we reformulate idle-mode measurement scheduling as a closed-loop control problem balancing two competing objectives: minimizing energy consumption and preserving reselection responsiveness. We design a prediction-based adaptive relaxation framework PARMA that (i) models short-term signal evolution, (ii) quantifies reselection risk relative to configured

thresholds, and (iii) dynamically adjusts measurement relaxation factors to maximize net utility. Unlike static relaxation rules, this approach selectively skips measurements when channel predictability is high and proactively reinstates full measurement frequency when risk increases.

Our evaluation demonstrates that predictive measurement control can achieve meaningful reductions in idle-mode energy consumption while maintaining mobility robustness. This case further reinforces the dissertation's central argument: measurement scheduling, like mobility and duplexing, can be reframed as a measurement-driven, cross-layer control problem governed by explicit system-level objectives.

1.5 Thesis Organization

This dissertation is structured to progressively develop and validate the central thesis that cellular resource management can be reformulated as a measurement-driven, closed-loop control problem with explicit system-level objectives. Chapter 2 provides background on cellular resource management mechanisms in 4G and 5G networks, including mobility management, duplex resource allocation, connectivity management, and idle-mode measurement procedures. It also introduces the cross-layer signals and control primitives that form the foundation for subsequent analysis and system design. Chapter 3 presents a large-scale empirical study of mobility management in operational 5G networks. It characterizes the performance and energy implications of frequent handovers and introduces a predictive framework that mitigates application-level degradation by anticipating mobility events. Chapter 4 examines duplex resource allocation in 5G. It demonstrates that static or reactive TDD policies are misaligned with dynamic uplink–downlink

asymmetry and presents a predictive, cross-layer control mechanism for adaptive slot configuration. Chapter 5 investigates connectivity management across heterogeneous cell deployments. It quantifies performance diversity across cell combinations and develops a centralized, policy-compliant framework that opportunistically selects configurations aligned with system-level objectives. Chapter 6 addresses idle-mode measurement scheduling. It reformulates measurement relaxation as a predictive control problem balancing reselection risk and energy efficiency. Finally, we summarize all related work in Chapter 7 before concluding the thesis in Chapter 8.

Chapter 2

Cellular Resource Management Foundations

Cellular resource management mechanisms in 4G and 5G networks are defined by detailed 3GPP specifications and shaped by operator-specific configuration policies. While Chapter 1 highlighted the limitations of existing rule-based approaches and the role of measurement-driven, cross-layer decision-making, this chapter establishes the technical foundations required to support that perspective. We begin with a concise overview of cellular architecture to situate subsequent control-plane procedures within their operational context. We then examine four representative control surfaces in contemporary networks: *(i)* mobility management, *(ii)* duplex resource allocation, *(iii)* connectivity management, and *(iv)* idle-mode measurement scheduling. For each, we describe the standardized procedures, decision triggers, and practical constraints under which they operate. The chapter concludes by identifying structural characteristics shared across these mechanisms, highlighting common design patterns and limitations that motivate the unified, cross-layer control formulation developed in later chapters.

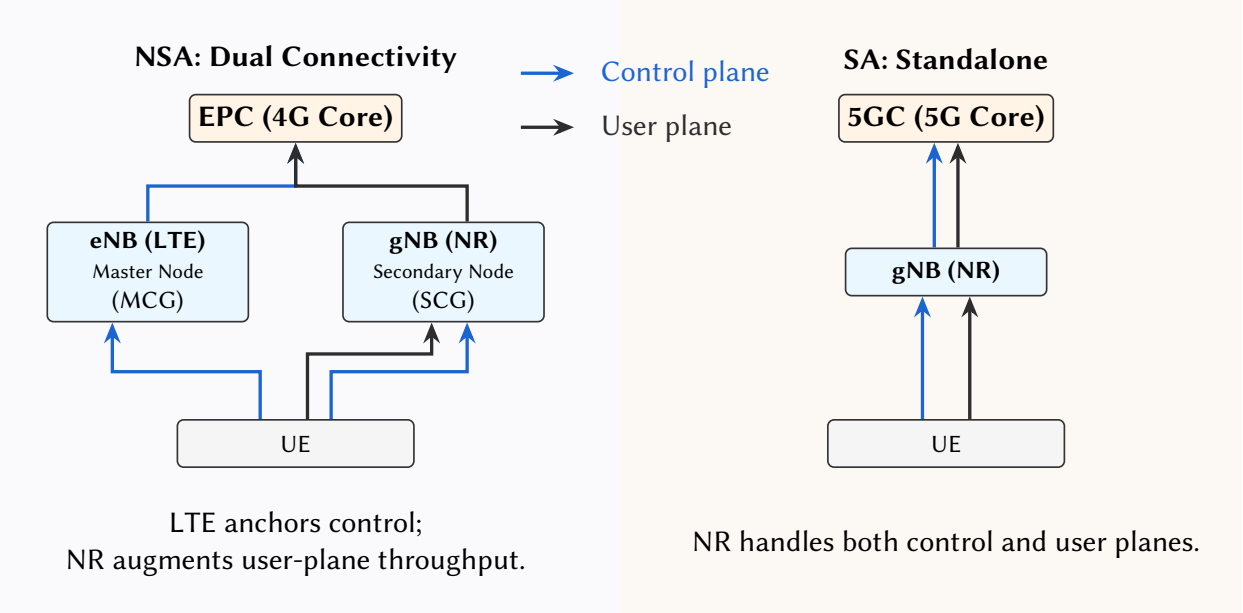


Figure 2.1: Simplified Radio Access Network (RAN) architecture under Non-Standalone (NSA) and Standalone (SA) deployments.

2.1 Cellular Architecture Overview

Modern cellular networks comprise two primary subsystems: the Radio Access Network (RAN) and the Core Network (CN). The RAN provides wireless connectivity between user equipment (UE) and the network, while the CN manages authentication, session establishment, mobility anchoring, and connectivity to external data networks. Although this dissertation focuses on resource management mechanisms within the RAN, understanding the overall architectural context is essential for interpreting subsequent control-plane procedures and system design choices.

Radio Access Network Architecture

In 4G LTE deployments, the RAN is composed of eNodeBs (eNBs), whereas in 5G New Radio (NR), the equivalent entity is the gNodeB (gNB). These base stations implement the cellular protocol

layers and are responsible for scheduling, radio resource allocation, measurement configuration, and mobility signaling with UEs. The RAN connects to the Core Network through standardized interfaces (S1 in LTE and NG in 5G), and neighboring base stations coordinate mobility through X2 (LTE) or Xn (5G) interfaces.

5G deployments operate in two primary architectural modes:

- **Non-Standalone (NSA):** 5G NR is deployed alongside LTE, where the LTE eNB anchors control-plane signaling and mobility procedures, while the NR gNB provides additional user-plane capacity through dual connectivity.
- **Standalone (SA):** 5G NR operates independently, with the gNB supporting both control-plane and user-plane functions and interfacing directly with the 5G Core (5GC).

Figure 2.1 illustrates these two deployment models. In NSA mode, the UE maintains simultaneous connections to an LTE eNB (master node) and an NR gNB (secondary node). Control-plane signaling flows primarily through the LTE anchor, while high-rate data is transmitted over NR bearers. In contrast, SA deployments eliminate LTE anchoring, allowing the UE to connect directly to the gNB for both control and user-plane communication. This architectural distinction introduces additional signaling complexity in NSA deployments and directly impacts mobility procedures, as handovers may involve coordination across both LTE and NR nodes.

Cellular Protocol Stack

Cellular communication is implemented through a layered protocol stack that separates user-plane data delivery from control-plane signaling. As illustrated in Figure 2.2, the *data plane* carries application traffic through the IP/transport layers and the 5G-specific Service Data Adaptation

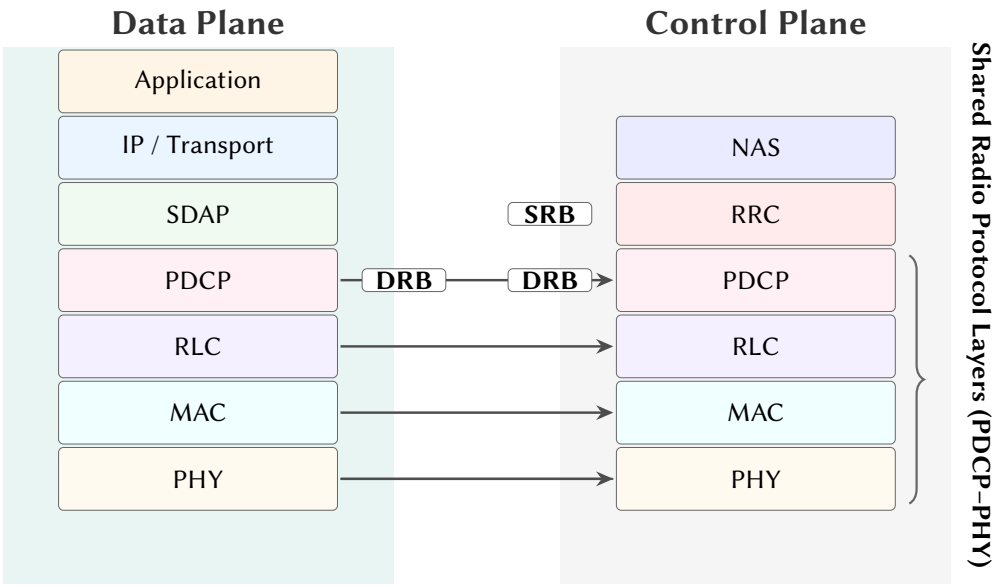


Figure 2.2: Simplified 5G NR protocol stack highlighting user/data plane and control plane layers.

Protocol (SDAP), followed by the shared radio protocol layers: Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), Medium Access Control (MAC), and the Physical (PHY) layer. In contrast, the *control plane* carries signaling messages through the Non-Access Stratum (NAS) and Radio Resource Control (RRC) layers, which configure radio resources, mobility procedures, and measurement operations. Both planes share the PDCP–PHY layers over the air interface, with data-plane traffic mapped to Data Radio Bearers (DRBs) and control-plane signaling mapped to Signaling Radio Bearers (SRBs). These protocol layers expose cross-layer signals—such as channel quality indicators from PHY, buffer occupancy from RLC/MAC, and measurement reports from RRC—that are later leveraged in measurement-driven resource management mechanisms.

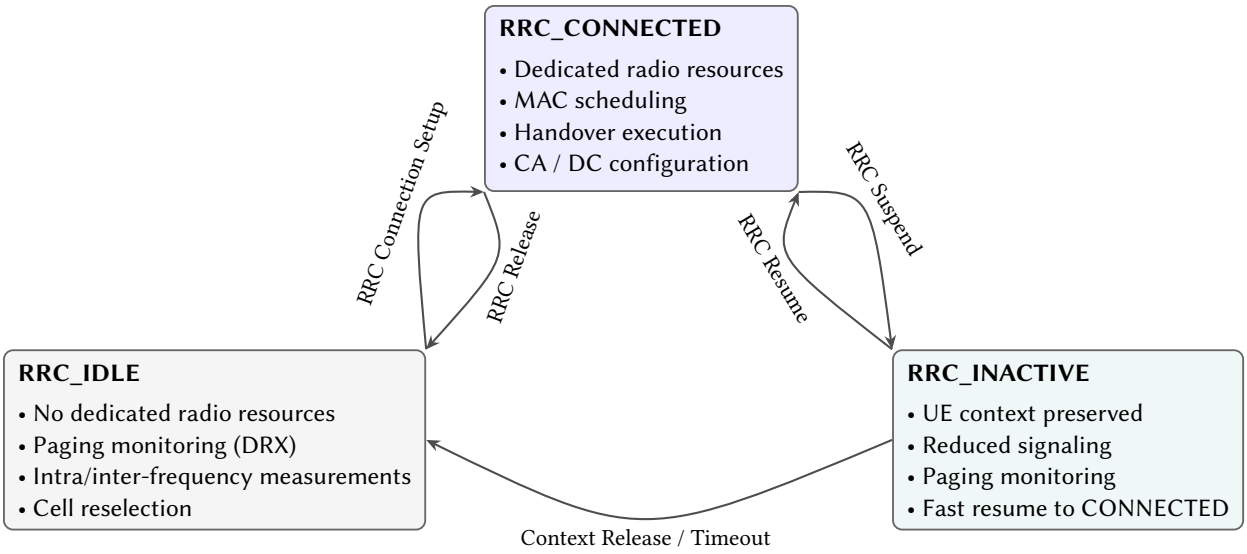


Figure 2.3: RRC state machine in 5G NR.

RRC States and Operational Context

UE behavior is governed by the Radio Resource Control (RRC) state machine, which determines resource allocation, measurement activity, and mobility procedures [12]. The primary RRC states in 5G NR are:

- **RRC_IDLE:** The UE is registered with the network but does not maintain dedicated radio resources. It periodically wakes to monitor paging occasions and performs measurements to support cell reselection.
- **RRC_CONNECTED:** The UE maintains active radio resources and participates in scheduling, data transmission, carrier aggregation, and handover procedures.
- **RRC_INACTIVE:** Introduced in 5G NR, this intermediate state preserves UE context to enable fast resume with reduced signaling overhead.

Figure 2.3 presents the simplified RRC state transition diagram. These states define the operational context in which resource management mechanisms operate. For example, connected-mode mobility is handled through handovers in RRC_CONNECTED, whereas cell reselection, paging monitoring, and measurement relaxation occur primarily in RRC_IDLE. This distinction will be important in later chapters that separately analyze connected-mode mobility and idle-mode measurement scheduling.

Cells, Component Carriers, and Multi-Cell Configuration

A single base station site may host multiple logical cells operating across different frequency bands and numerologies. Each cell corresponds to a component carrier (CC), defined as a contiguous frequency block with its own physical cell identity and configuration parameters [1]. Consequently, a UE at a given location may observe multiple candidate cells across low-, mid-, and high-band spectrum.

5G NR supports simultaneous connectivity to multiple serving cells through Carrier Aggregation (CA) and Dual Connectivity (DC). At minimum, the UE maintains a *Primary Cell (PCell)* that anchors control signaling. Additional *Secondary Cells (SCells)* may be configured to increase bandwidth and data throughput. In NSA deployments, LTE and NR cells are organized into a *Master Cell Group (MCG)* and *Secondary Cell Group (SCG)*, with a *Primary Secondary Cell (PSCell)* anchoring NR control-plane signaling within the SCG.

2.2 Mobility Management in 4G and 5G Networks

Mobility management ensures service continuity as UE moves across cellular coverage areas. Since cells provide limited geographic coverage, ongoing sessions must be transferred between serving cells to maintain connectivity and acceptable radio conditions. Mobility procedures therefore operate as a feedback process in which the network monitors radio measurements and reconfigures the UE's serving cell when predefined criteria are met.

Traditionally, mobility management has prioritized seamless connectivity and link reliability. Standardized decision rules rely primarily on radio link quality indicators, such as reference signal strength and quality, to prevent radio link failure and preserve coverage. While effective for maintaining connectivity, these heuristics do not explicitly account for higher-layer performance objectives such as throughput stability, latency, or energy efficiency.

Mobility events affect both control-plane signaling and user-plane performance. Handover preparation and execution introduce signaling overhead and may temporarily disrupt data delivery through buffering, packet loss, or reordering. As 5G deployments introduce heterogeneous bands, dense cell layouts, and dual connectivity, the frequency and complexity of mobility events increase, amplifying their impact on application performance.

Mobility Context Across RRC States

Mobility procedures operate differently depending on the UE's RRC state. In RRC_CONNECTED, mobility is handled through handovers, where the network transfers the UE's active connection from a source cell to a target cell while preserving ongoing data sessions. This process involves

Table 2.1: LTE/NR measurement reporting events used for mobility decisions. M_S and M_N denote measurements of the serving and neighboring cells, respectively.

Event	Description	Trigger Condition
A1	Serving cell becomes better than a threshold	$M_S > \Phi_{A1}$
A2	Serving cell becomes worse than a threshold	$M_S < \Phi_{A2}$
A3	Neighboring cell becomes offset better than serving cell	$M_N > M_S + \Delta_{A3}$
A4	Neighboring cell becomes better than a threshold	$M_N > \Phi_{A4}$
A5	Serving becomes worse than threshold 1 and neighbor becomes better than threshold 2	$M_S < \Phi_{A5}^{(1)}$ $M_N > \Phi_{A5}^{(2)}$
B1	Inter-RAT neighbor becomes better than a threshold	$M_N > \Phi_{B1}$
Periodic	Periodic reporting of measurement values	N/A

coordinated control-plane signaling, resource preparation at the target cell, and a brief interruption of user-plane traffic during execution. Connected-mode mobility therefore directly affects application performance through transient latency spikes, packet loss, or throughput fluctuations.

In contrast, mobility in RRC_IDLE and RRC_INACTIVE is managed through cell reselection, where the UE autonomously selects a new serving cell based on broadcast configuration and locally measured signal conditions. Idle-mode mobility does not interrupt active data sessions but requires periodic measurements and paging monitoring, contributing to control-plane signaling and UE energy consumption. Together, connected-mode handovers and idle-mode reselection form complementary mechanisms that maintain connectivity across varying activity levels and mobility patterns.

Measurement Framework for Mobility Decisions

Mobility decisions in cellular networks are driven by standardized measurement procedures that enable the network to observe radio conditions and identify candidate target cells. UEs continuously monitor signal quality using physical-layer metrics such as Reference Signal Received

Power (RSRP), Reference Signal Received Quality (RSRQ), and Signal-to-Interference-plus-Noise Ratio (SINR), as defined in 3GPP specifications [9]. These measurements are filtered at the UE and reported to the serving cell through Radio Resource Control (RRC) signaling [12]. In addition to signal strength indicators, Channel Quality Indicator (CQI) feedback derived at the PHY/MAC layers supports scheduling decisions and indirectly influences mobility behavior.

Measurement behavior is governed by configuration parameters provided by the network, including measurement objects, reporting configurations, and measurement identities [12]. Event-based reporting is widely used in practice, where the UE generates a measurement report when predefined conditions are satisfied. Table 2.1 summarizes the commonly used LTE/NR measurement events and their trigger conditions. Events such as A3 (neighbor becomes better than serving cell by an offset) are widely used for intra-frequency handovers, while A2/A5 provide mechanisms for threshold-based and conditional handover triggering. Additional parameters such as hysteresis margins, time-to-trigger (TTT), and measurement filtering coefficients control reporting stability and prevent rapid oscillations between cells [12].

The measurement pipeline therefore consists of configuration, continuous monitoring, filtering, and event-triggered reporting. These reports form the primary input to mobility decision logic at the serving base station, which evaluates candidate cells and determines whether a handover should be initiated. Understanding this standardized measurement framework is essential for interpreting the timing and behavior of mobility events observed in operational networks.

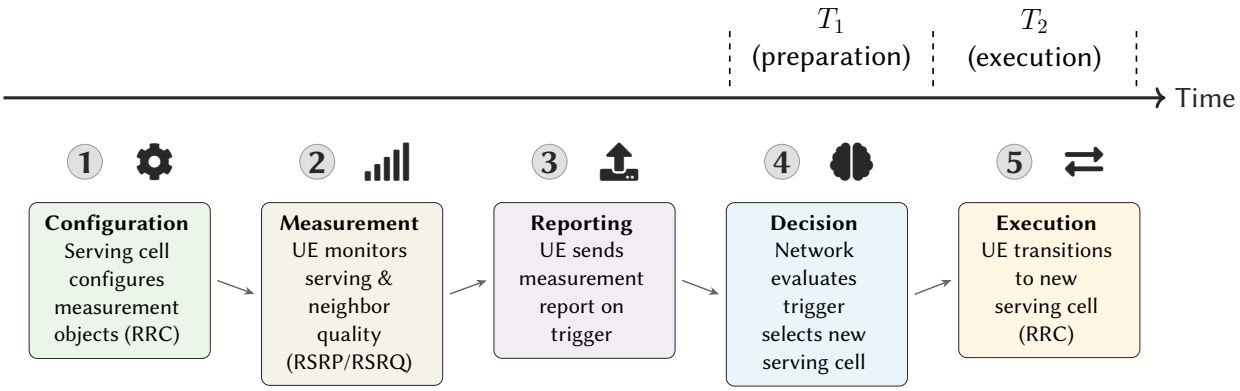


Figure 2.4: Logical phases of a connected-mode handover in 5G.

Handover Procedure

Mobility support is a fundamental service in cellular networks, and handover (HO) is the primary mechanism that enables connected-mode mobility. Figure 2.4 illustrates a simplified handover procedure consisting of five logical phases. These phases reflect the standardized measurement-driven control loop defined by 3GPP specifications [12, 1].

In step ①, the serving (primary) cell configures the UE with handover-related measurement parameters, including thresholds, offsets, and reporting rules via RRC signaling [12]. In step ②, the UE continuously monitors the radio quality of serving and neighboring cells according to the configured measurement objects. When a configured measurement event is satisfied, an event is raised at the UE and summarized in a measurement report, which is transmitted to the serving cell in step ③. Common event types and their trigger conditions are summarized in Table 2.1.

Upon receiving the measurement report, the serving cell evaluates candidate target cells and makes a handover decision in step ④. If a suitable target cell is selected, the serving cell initiates handover preparation by requesting the target cell to allocate radio resources for the incoming UE over the Xn/X2 interface [1]. We refer to this interval as the *handover preparation stage* (T_1). In

Table 2.2: Classification of handover procedures observed in LTE, NSA 5G, and SA 5G.

Procedure Type	RAT Transition	HO Category	Acronym
SCG Addition	$4G \rightarrow 5G$	5G (NSA)	SCGA
SCG Release	$5G \rightarrow 4G$	5G (NSA)	SCGR
SCG Modification	$5G \rightarrow 5G$	5G (NSA)	SCGM
SCG Change	$5G \rightarrow 4G \rightarrow 5G$	5G (NSA)	SCGC
MeNB HO	$5G \rightarrow 5G$	4G (NSA anchor)	MNBH
MCG HO (SA)	$5G \rightarrow 5G$	5G (SA)	MCGH
LTE HO (NSA)	$5G \rightarrow 5G$	4G (NSA)	LTEH
LTE HO (LTE)	$4G \rightarrow 4G$	4G (LTE)	LTEH

step ⑤, the serving cell transmits an RRC reconfiguration message instructing the UE to switch to the target cell. The UE synchronizes with the target cell, completes the Random Access (RACH) procedure, and responds with an RRC reconfiguration complete message [11]. This interval corresponds to the *handover execution stage* (T_2), during which user-plane traffic may be temporarily interrupted.

While the logical handover procedure is standardized, the specific signaling sequences and mobility behaviors differ across LTE, NSA 5G, and SA 5G deployments. We next summarize the taxonomy of handover procedures observed in these architectures.

Handover Types in LTE and 5G

The classification of handovers has become increasingly complex with the introduction of dual connectivity and heterogeneous radio access technologies in 5G. Table 2.2 summarizes the handover procedures observed in LTE, Non-Standalone (NSA) 5G, and Standalone (SA) 5G deployments, along with the terminology used throughout this dissertation. Figure 2.5 illustrates the SCG mobility procedures used in NSA 5G to add, modify, and release NR cells.

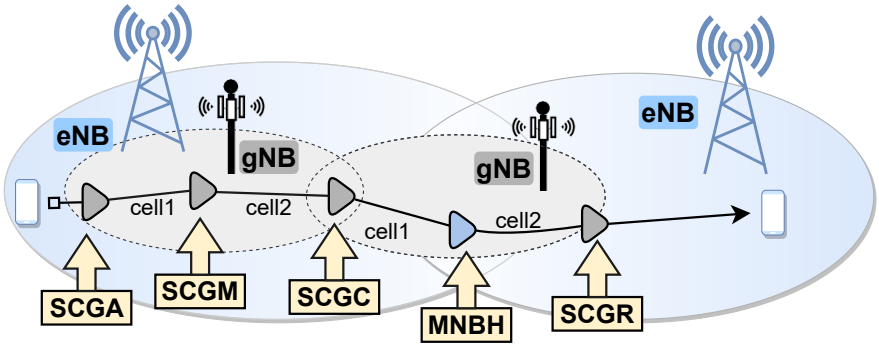


Figure 2.5: SCG HO procedures for mobility in NSA 5G.

In NSA deployments, LTE acts as the control-plane anchor, and cells associated with the eNB form the *Master Cell Group* (MCG), while NR cells connected to the gNB form the *Secondary Cell Group* (SCG) [6]. Consequently, mobility procedures in NSA 5G include both LTE-based handovers and NR-specific SCG procedures. *SCG Addition* introduces NR resources to an existing LTE connection, whereas *SCG Release* removes NR connectivity. *SCG Modification* switches between NR cells within the same gNB, while *SCG Change* performs an indirect transition between gNBs through a release-and-add sequence, since direct NR-to-NR handover is not supported in NSA. A *MeNB HO* changes the LTE anchor cell while maintaining the NR secondary connection.

In SA deployments, the LTE anchor is removed, and mobility is handled through *MCG handovers*, where the UE transitions directly between NR cells. In addition, LTE handovers may still occur in NSA deployments when the LTE anchor cell changes, even if the NR secondary cell remains unchanged. These diverse mobility procedures reflect the architectural coupling between LTE and NR in NSA networks and contribute to the increased frequency and complexity of handover events observed in operational 5G systems.

Architectural Factors Increasing Mobility Complexity in 5G

Mobility management in 5G is significantly more complex than in previous cellular generations due to architectural and spectrum-level changes. Modern deployments operate across heterogeneous frequency bands, including low-band, mid-band, and millimeter-wave spectrum, each with different coverage characteristics and propagation behavior. Higher-frequency cells typically provide smaller coverage footprints, leading to more frequent cell transitions during mobility. In addition, dense small-cell deployments and beamforming-based transmission further increase the granularity of mobility events, introducing not only cell-level but also beam-level transitions that can affect radio link stability.

The coexistence of LTE and NR in NSA architectures further complicates mobility behavior. In NSA deployments, mobility may involve coordinated procedures across both LTE anchor cells and NR secondary cells, resulting in multiple signaling exchanges and indirect transitions between gNBs. These factors increase both the frequency and diversity of handover events compared to LTE-only systems. As a result, mobility procedures in operational 5G networks can introduce greater signaling overhead, transient performance degradation, and energy consumption, motivating the empirical characterization and analysis presented in the following chapter.

2.3 Duplex Resource Allocation

5G Frame Structure. 5G New Radio (NR) introduces a flexible frame structure designed to support diverse service requirements. Unlike LTE, which uses a fixed numerology, 5G supports multiple numerologies indexed by $\mu \in [0, 4]$, corresponding to subcarrier spacings of $2^{\mu} \cdot 15$ kHz. As shown in Figure 2.6, the frame hierarchy consists of 10 ms radio frames, each divided into ten

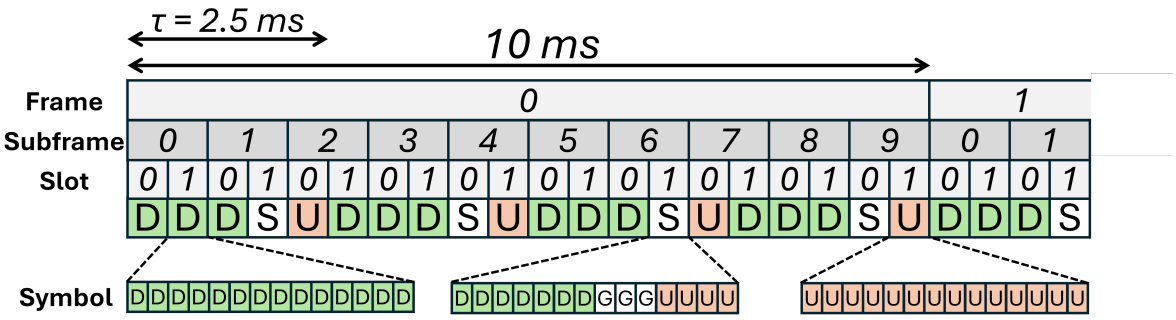


Figure 2.6: 5G frame structure and TDD pattern for numerology $\mu = 1$.

1 ms subframes. Each subframe contains 2^{μ} slots, and each slot has a duration of $2^{-\mu}$ ms. With a normal cyclic prefix, each slot contains 14 OFDM symbols, which form the smallest scheduling unit in NR.

5G NR TDD. Time Division Duplexing (TDD) allows uplink (UL) and downlink (DL) transmissions to share the same spectrum while separating them in time. In practice, most mid-band and millimeter-wave deployments use TDD to enable flexible spectrum utilization and adapt to asymmetric traffic demands [15]. 5G supports three modes of TDD operation: static TDD with fixed UL/DL allocation, semi-static TDD configured through higher-layer signaling, and dynamic TDD, which adapts slot allocation based on real-time network conditions. Dynamic TDD subsumes the other two approaches as special cases, and is the focus of this work.

TDD Policy. A TDD policy defines the allocation of slots and symbols to UL, DL, and guard periods within a repeating transmission pattern. The periodicity τ specifies how frequently the pattern repeats. For a given numerology μ, a pattern contains $T_s = 2^{\mu} \cdot \tau$ slots. Guard periods are inserted to prevent cross-link interference and to account for propagation delays [32]. While 3GPP specifies the signaling mechanisms that communicate slot configuration to UEs, the algorithm used by the base station (BS) to determine the TDD policy is left open-ended. This

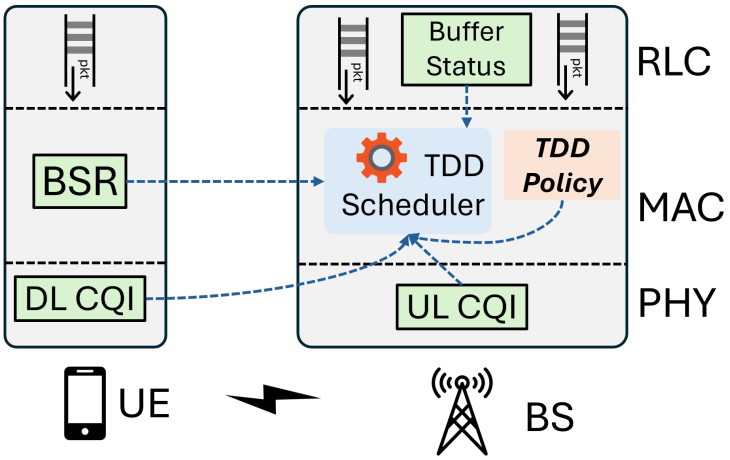


Figure 2.7: Overview of radio resource scheduling in 5G TDD networks.

flexibility enables operators to tailor policies to traffic conditions but also introduces significant design complexity.

TDD Resource Scheduling. Figure 2.7 illustrates the interaction between TDD policy selection and radio resource scheduling. UEs periodically report channel quality through Channel Quality Indicator (CQI) measurements, while UL queue backlog is conveyed via Buffer Status Reports (BSRs). Based on CQI, queue backlog, and the configured TDD pattern, the MAC scheduler assigns OFDM symbols to UEs and selects the modulation and coding scheme (MCS), which determines the transport block size and achievable data rate. Because TDD policy determines the available UL and DL symbols, it directly influences scheduling decisions and, consequently, application performance.

Interference and Guard Period Considerations. In TDD systems, UL and DL transmissions occur in adjacent time intervals on the same frequency band. Without sufficient separation, simultaneous UL and DL transmissions across neighboring cells can lead to cross-link interference, degrading signal quality and reducing achievable throughput. Guard periods are therefore inserted between UL and DL symbols to allow for radio switching time and propagation delays.

While necessary for reliable operation, guard periods reduce the number of symbols available for data transmission, introducing a trade-off between robustness and spectral efficiency. The duration and placement of guard symbols thus play a critical role in determining the effectiveness of a TDD policy.

Traffic Load and Base Station Objectives. A base station's traffic load is commonly characterized by the amount of UL and DL data waiting in transmission queues. This definition reflects the instantaneous demand placed on the scheduler and directly affects resource allocation decisions. In contrast, base station throughput measures the rate at which data is transmitted and received over the air interface. In practice, TDD policies must balance multiple objectives: maximizing aggregate throughput, minimizing latency (particularly for UL traffic), and maintaining fairness across UEs. Because application-level QoE is not directly observable at the base station, these objectives are typically expressed in terms of radio-layer Quality of Service (QoS) metrics such as queue delay, scheduling latency, and achieved throughput.

Practical Constraints in Dynamic TDD. Although dynamic TDD provides flexibility, frequent changes to slot configuration introduce practical challenges. Updating the TDD pattern requires coordination with UEs through control signaling, and abrupt policy shifts may disrupt Hybrid ARQ processes or interfere with transport-layer congestion control and application-layer rate adaptation. Furthermore, the UL/DL asymmetry in wireless channels—where UL often experiences lower transmit power and higher latency—requires careful consideration when reallocating slots. Effective TDD adaptation must therefore balance responsiveness to traffic dynamics with stability to avoid unintended cross-layer interactions.

2.4 Multi-Cell Connectivity and CM Procedures

Cells and Multi-Cell Operation. At any geographic location, a UE is typically within the coverage of multiple cells, each operating over a continuous frequency block referred to as a *component carrier* (CC). A Base Station (BS)—called an eNodeB in 4G and a gNodeB in 5G—hosts the physical infrastructure for one or more such cells. Modern cellular networks are inherently multi-cell and multi-band, spanning Low-Band, Mid-Band, and High-Band frequencies.

A key feature of LTE and 5G is Carrier Aggregation (CA), which combines multiple cells to increase data throughput. In 5G NSA deployments, Dual Connectivity (DC) extends CA by allowing a UE to simultaneously connect to both LTE and NR cells. Serving cells include a mandatory *Primary Cell (PCell)*, responsible for control-plane signaling and data access, and optional *Secondary Cells (SCells)* that provide additional data-plane capacity. In NSA-5G, the *Primary Secondary Cell (PSCell)* anchors 5G signaling in addition to the LTE PCell.

We use the term *cell combination* to denote the set of serving cells configured for a UE, consisting of a mandatory PCell and optional PSCell and/or SCells. A cell combination is considered *unique* if it differs in any of its serving cells (PCell, PSCell, or SCells). As heterogeneous deployments proliferate, the number of feasible cell combinations at a given location can be substantial.

Figure 2.8 illustrates a representative multi-cell configuration. The UE is connected to one PCell and multiple SCells that together form the active serving set. Several additional neighboring cells are available but not currently configured. Different combinations of serving cells can yield distinct performance characteristics in terms of throughput, latency, and reliability. Selecting among these combinations is the central problem addressed by connectivity management.

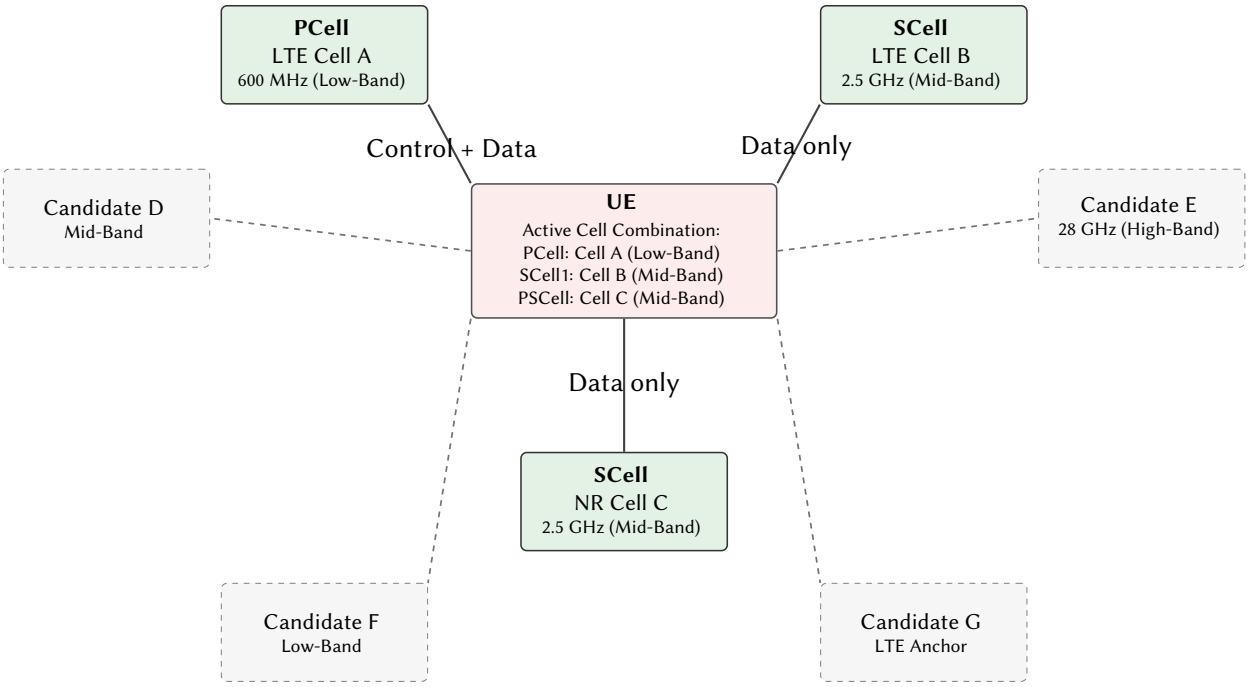


Figure 2.8: Example of a UE configured with a Primary Cell (PCell) and multiple Secondary Cells (SCells) under carrier aggregation.

Connectivity Management (CM) Procedures. Connectivity Management (CM) procedures determine which serving cells a UE attaches to at any moment. These procedures include:

- **Cell selection:** choosing an initial serving cell when the UE first accesses the network.
- **Cell reselection:** switching serving cells while in idle or inactive state.
- **Handover:** transferring an active connection between cells to maintain service continuity.
- **Load balancing:** redistributing traffic across cells to prevent congestion.
- **CA/DC:** adding or removing SCells or PSCells to adjust aggregate bandwidth.

Although these procedures differ in trigger conditions and UE state (idle, inactive, connected), they all perform variations of three primitive actions: *adding*, *modifying*, or *removing* serving

Table 2.3: An overview of common CM procedures and their UE data transmission modes (idle, inactive, connected), actions (add, modify, remove cells), and criteria (cell accessibility, link quality, absolute priority).

Procedure	Action on UE's Cell Set	Mode	Cell Accessibility	Radio Link Quality	Absolute Priority
Cell Selection	Adds PCell	Idle/Inactive	Mandatory	Irrelevant	Primary
Cell Reselection	Modify PCell	Idle/Inactive	Mandatory	Primary	Primary
Handover	Modify PCell/PSCell	Connected	Mandatory	Primary	Secondary
Carrier Aggregation	Add/Remove SCells	Connected	Mandatory	Primary	Primary
Dual Connectivity	Add/Remove PSCell	Connected	Mandatory	Primary	Secondary
Load Balancing	Modify PCell/PSCell	Connected	Mandatory	Irrelevant	Primary

cells. Legacy CM schemes typically rely on radio link quality metrics (*e.g.,* RSRP, RSRQ), absolute priorities, and accessibility constraints. Table 2.3 summarizes these legacy CM schemes and highlights that they are largely driven by link quality rather than application-level performance objectives.

Terminology for Cell Combinations. For brevity, we adopt the following notation to describe cell combinations:

- 61^5(3350) denotes a UE connected to a 5G PCell with Physical Cell ID (PCI) 61 operating at 3350 MHz.
- 85^4(1850)/61^5(3350) represents an NSA-5G UE with LTE PCell 85 and 5G PSCell 61.
- 85^4(1850)/61^5(3350)/{108^5(3370), 111^5(3330)} indicates the addition of two 5G SCells (108 and 111) to the previous configuration.

This notation allows us to concisely describe complex CA/DC configurations and reason about the space of available cell combinations. As we show later in §5.2, the diversity of accessible combinations creates substantial performance opportunities that legacy CM schemes do not exploit.

2.5 Idle-State RRM Measurements

In 5G NR, as in previous cellular generations, a UE transitions among RRC states depending on activity (see §2.1). In RRC_CONNECTED, the UE maintains an active radio link for data transmission and reception. In RRC_IDLE, the UE has no active data session but remains registered to the network, enabling fast service resumption upon incoming traffic. 5G further introduces RRC_INACTIVE, an intermediate state where UE context is preserved in the core network to reduce reconnection latency while limiting signaling overhead.

Although idle and inactive states eliminate continuous data-plane activity, the UE must still remain aware of the surrounding radio environment. This awareness is maintained through periodic paging monitoring and radio resource management (RRM) measurements, which together dominate idle-mode energy consumption.

Idle and Inactive State Measurements

Even in idle or inactive states, the UE continuously evaluates whether it should remain camped on the current cell or perform cell reselection. Cell reselection ensures mobility continuity when radio conditions change. Idle-mode measurement procedures include: **(i) Camped-cell monitoring:** measuring metrics such as RSRP and RSRQ of the serving cell. **(ii) Intra-frequency neighbor measurements:** monitoring cells operating on the same carrier frequency. **(iii) Inter-frequency neighbor measurements:** scanning cells operating on different carrier frequencies.

Measurement configuration parameters—including frequency priorities, thresholds, hysteresis offsets, and timing—are broadcast by the network through System Information Blocks (SIBs). The procedures are governed primarily by 3GPP TS 38.304 [10].

Inter-frequency measurements incur higher overhead than intra-frequency measurements because the UE must retune its RF front-end, switch carriers, and wait for measurement opportunities. These operations extend the receiver active duration and increase power draw.

Paging and DRX Operation

Paging is the mechanism by which the network notifies an idle UE of incoming data or signaling events. Paging occasions (POs) occur periodically according to a network-configured paging cycle [10]. A commonly observed paging cycle in commercial networks is 1.28 seconds.

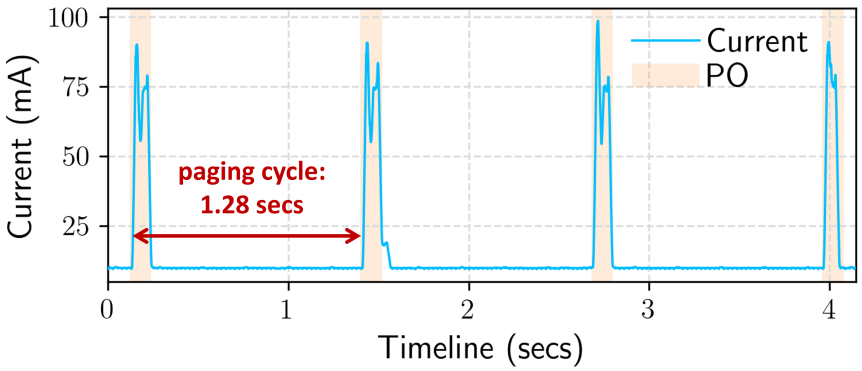


Figure 2.9: Illustration of a Paging Occasion (PO) in 5G NR. The UE periodically wakes to monitor paging messages and perform configured measurements, leading to current spikes.

Between paging occasions, the UE remains in a low-power Discontinuous Reception (DRX) sleep state. During each PO, the UE: (i) Activates its RF receiver and baseband, (ii) Monitors the Physical Downlink Control Channel (PDCCH), (iii) Decodes the paging message if present, and (iv) Performs configured RRM measurements.

Figure 2.9 illustrates the periodic wake-up behavior. Each PO introduces a short burst of current consumption. When inter-frequency measurements are configured, the active window lengthens due to carrier retuning and measurement gaps, further increasing energy cost.

The total idle-mode energy consumption is therefore primarily determined by paging cycle periodicity, duration of each active window, and measurement configuration complexity (intra- vs inter-frequency scans). Figure 2.10 further zooms into the idle-state PO.

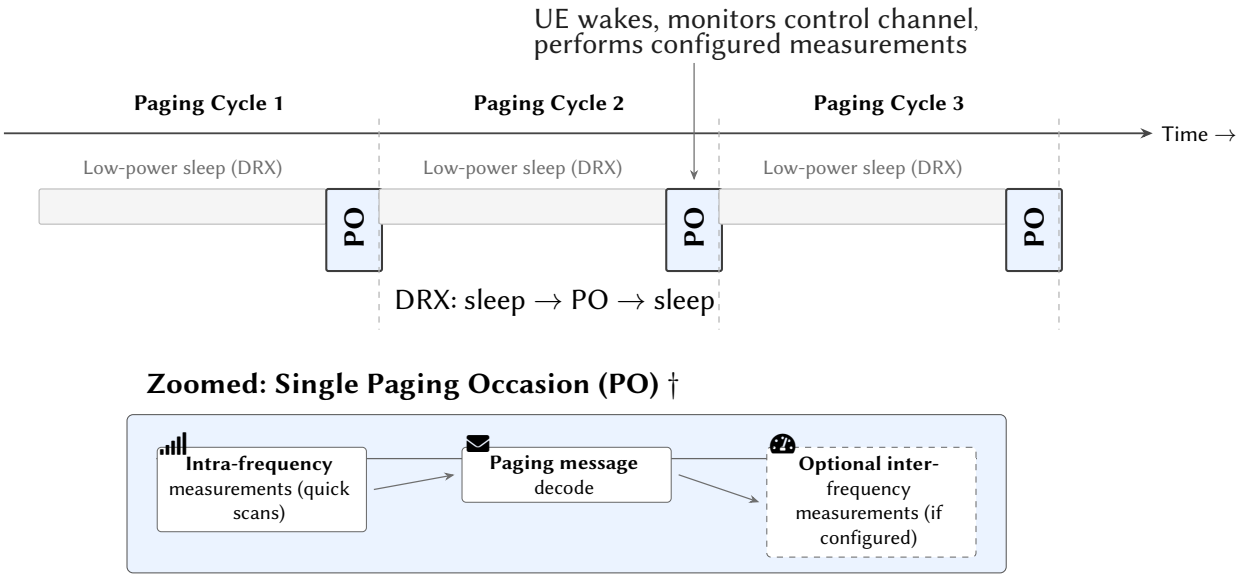


Figure 2.10: Zooming into the idle-state 5G NR paging occasion.

Measurement Scheduling in Idle Mode

By default, idle-mode measurements are performed at every paging occasion. That is, each PO may trigger camped-cell evaluation, intra-frequency scans, and inter-frequency scans (if configured). This default behavior corresponds to a measurement relaxation factor $R = 1$, meaning no relaxation. As a result, the UE may perform repeated inter-frequency measurements even when the serving-cell signal remains stable for long periods. Figure 2.11 illustrates this default scheduling behavior.

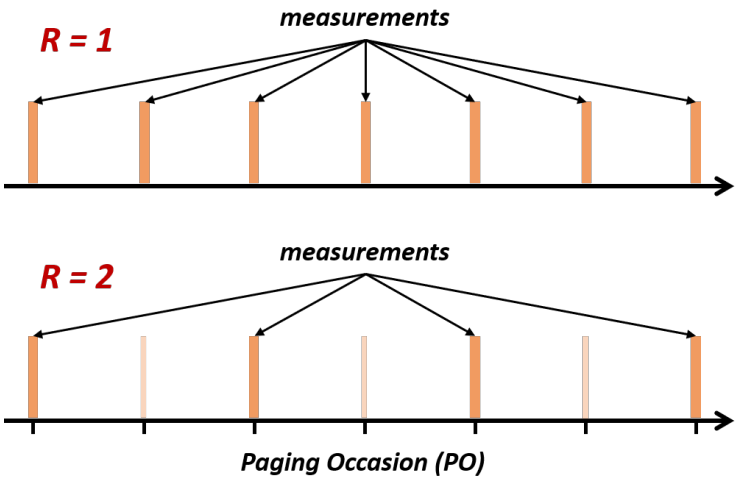


Figure 2.11: Measurement scheduling in idle mode. Top: default operation ($R = 1$) where measurements occur at every PO. Bottom: relaxed operation ($R = 2$) where measurements occur less frequently.

3GPP Measurement Relaxation

To reduce idle-mode energy consumption, 3GPP introduces *measurement relaxation* mechanisms under specific conditions [10]. Relaxation aims to suppress unnecessary neighbor-cell measurements when the probability of reselection is low.

Two primary conditions enable relaxation:

1. **Low mobility condition:** The UE remains camped on the same cell with stable signal quality for an extended period.

2. **Not-cell-edge condition:** The serving cell's signal strength and quality exceed defined thresholds, indicating low reselection risk.

When relaxation is triggered, the UE applies a relaxation factor R:

- $R = 1$: No relaxation (measure every PO).

- $R = 2$: Measure every other PO.

- Higher R: Skip additional measurement opportunities.

The relaxation factor depends on parameters such as frequency range, UE power class, DRX cycle length, and network configuration [7]. While this mechanism reduces energy overhead, it is inherently heuristic and limited to mobility- and signal-strength-based triggers. It does not exploit temporal predictability of channel evolution beyond the standardized criteria.

Energy Implications

Idle-mode energy consumption is dominated not by continuous transmission, but by repeated receiver wake-ups and measurement activity. Even small extensions of the active window during each PO accumulate over thousands of cycles per day. Inter-frequency scans are particularly expensive because they require RF retuning and measurement gaps. Therefore, reducing unnecessary measurement executions—even modestly—can translate into measurable battery savings over extended idle periods. However, overly aggressive suppression risks delayed cell reselection and degraded mobility robustness. This tension between energy savings and reselection reliability motivates the need for a prediction-aware, risk-controlled measurement relaxation framework, which we develop in chapter §6.

2.6 Structural Characteristics of Current Mechanisms

The preceding sections described mobility management, duplex resource allocation, connectivity management, and idle-mode measurement scheduling as specified in contemporary 4G and 5G systems. Although these mechanisms operate at different layers and RRC states, they share several structural characteristics that are relevant to this dissertation.

First, decision-making is predominantly threshold-based and rule-driven. Mobility triggers rely on signal comparisons with configured offsets and timers [12]. Cell reselection is governed by frequency priorities and S-criteria [10]. TDD configurations are often static or semi-static, with limited dynamic adaptation [3]. Measurement relaxation applies under predefined signal stability conditions [10]. In each case, actions are triggered when specific radio-layer metrics cross configured boundaries.

Second, operation is largely reactive rather than predictive. Handover decisions are initiated after measurement events indicate signal degradation. Duplex reconfiguration, when present, typically responds to observed traffic conditions. Connectivity changes follow threshold satisfaction rather than anticipatory evaluation. Idle-mode measurement scheduling reduces frequency only after coarse stability conditions are satisfied. The control logic does not explicitly anticipate short-term system evolution.

Third, resource management objectives are typically implicit and radio-centric. The standardized procedures prioritize link continuity, coverage robustness, and interference mitigation. While these objectives are essential, they are not explicitly formulated in terms of system-level performance metrics such as application throughput stability, end-to-end latency, fairness across heterogeneous cell combinations, or UE energy efficiency. Higher-layer performance effects are indirect consequences rather than explicit optimization targets.

Fourth, control surfaces operate with limited coordination across layers. Mobility management, duplex configuration, connectivity management, and measurement scheduling are specified and implemented largely independently. Although modern networks expose rich cross-layer observability—including PHY-layer signal metrics, MAC-layer buffer states, RRC signaling context, transport-layer congestion signals, and application-layer behavior—these signals are not

systematically integrated into a unified decision framework. Decisions at one control surface may influence performance at another, but coordination is not structurally enforced.

Finally, the policy logic governing these mechanisms is configurable but fragmented. Operators adjust thresholds, hysteresis values, priorities, and timing parameters to suit deployment goals. However, these configurations are often tuned per mechanism rather than optimized jointly across control surfaces. As a result, system behavior emerges from the interaction of independently configured procedures.

These structural characteristics establish the baseline against which this dissertation is positioned. The following chapters examine each control surface empirically and demonstrate that these mechanisms can be treated as measurement-driven, closed-loop control problems that incorporate cross-layer signals and explicitly optimize system-level objectives. The goal is not to replace standardized procedures, but to augment their decision logic in a principled and deployment-compatible manner.

Chapter 3

Measurement, Modeling, and Mitigation of 5G Handovers

3.1 Introduction

Mobility management in 5G has become increasingly complex due to heterogeneous spectrum usage, dense small-cell deployments, and the coexistence of LTE and NR through NSA and SA architectures. These factors lead to more frequent and diverse handover (HO) events compared to prior cellular generations. While handovers are essential for maintaining connectivity, prior studies in LTE and emerging 5G deployments show that frequent mobility events can introduce throughput fluctuations, latency spikes, and, in extreme cases, temporary service disruptions [91, 177, 184, 49, 116, 182, 117, 122, 121]. Such transient disruptions can significantly degrade the QoE of modern latency-sensitive applications, motivating a deeper empirical understanding of mobility behavior in operational 5G networks.

Study Goal, Challenges, and Data Collection. (§3.2) Given the importance and complexity of 5G HOs, it is critical to understand how mobility management operates in commercial deployments. To this end, we conduct a comprehensive measurement-driven study of 5G mobility management in operational networks. Unlike controlled laboratory experiments, measuring

5G HOs in the wild presents several challenges: obtaining control-plane signaling events from unrooted smartphones, surveying diverse 5G architectures and bands across multiple carriers, coordinating cross-layer data collection, and accurately profiling the energy impact of mobility events.

To address these challenges, we develop a measurement platform consisting of: (i) multiple 5G smartphones connected to three major U.S. carriers; (ii) custom software that captures mobility-related information from unrooted devices; (iii) a professional diagnostic tool that records cellular control-plane events; and (iv) a physical power monitor with an external battery pack for accurate energy profiling. Using this platform, we conduct a cross-country drive test covering more than 6,200 km, including both highway and urban measurements. Across the campaign, we collect over 600 GB of logs and observe more than 47,000 mobility procedures spanning multiple dimensions: carriers (Verizon, T-Mobile, AT&T), radio technologies (4G and 5G), deployment architectures (NSA and SA), and frequency bands (low-band, mid-band, and mmWave). To our knowledge, this constitutes one of the largest cross-layer mobility measurement datasets collected from commercial 5G networks.

Leveraging this dataset (Table 3.1), we conduct a detailed analysis of 5G HOs and their performance implications. Our findings reveal significant disparities in mobility behavior across carriers, architectures, and bands, with measurable impact on application performance, control-plane signaling, and UE energy consumption.

How do 5G HOs Impact Applications? (§3.3) To quantify the application-level impact of mobility, we evaluate three representative workloads: live video conferencing, real-time volumetric video streaming, and cloud gaming. Our experiments show that HO events introduce noticeable QoE degradation. For example, during video conferencing, the average frame loss rate

Table 3.1: Dataset overview: network footprint, mobility events, and trace coverage.

	Verizon	**T-Mobile**	**AT&T**
Network footprint			
# Unique cells (*i.e.,* towers)	3,030	5,535	3,544
# 5G-NR bands	4	2	4
# LTE bands	5	9	6
Drive coverage			
City distance (4 major cities)	697 km	712 km	652 km
Inter-state distance (freeways)	4,855 km	5,560 km	4,855 km
Mobility events			
4G/LTE handovers	7,001	9,500	7,491
5G-NSA mobility procedures	4,611	11,107	6,880
5G-SA handovers	–	465	–
Trace duration			
5G-NR Low-band traces	723 min	1,532 min	1,063 min
5G-NR Mid-band traces	15 min	1,088 min	132 min
5G-NR mmWave traces	258 min	–	172 min
5G-NSA traces	996 min	2,204 min	1,366 min
5G-SA traces	–	416 min	–
4G/LTE traces	2,412 min	1,510 min	2,038 min

increases by 2.24× and end-to-end latency increases by 2.26× on average (up to 14.5×). In 4K cloud gaming at 60 FPS, we observe a 3.64× increase in dropped frames during HO events. These results indicate that mobility-induced disruptions can significantly degrade application performance even when handovers are triggered by improved signal strength.

The severity of performance degradation depends on HO type, radio band, and architecture. In NSA deployments, where LTE provides the control plane and NR carries high-speed data (denoted as *NSA-4C*), separate HOs occur over LTE and NR nodes, leading to increased HO frequency. In particular, mmWave deployments experience more frequent and disruptive mobility events due to smaller coverage regions and directional beam management. On the other hand,

applications using *dual connectivity*—where traffic can be served simultaneously over LTE and NR—can partially mitigate HO-induced disruptions through multi-radio diversity.

What are the Key Characteristics of 5G HOs? (§3.4) Motivated by the application-level findings, we conduct an in-depth characterization of HO behavior, focusing on HO frequency, duration, and UE energy consumption. We observe that 5G HOs occur more frequently than LTE HOs; for example, during freeway driving, 5G mobility events occur approximately every 0.4 km on average, compared to 0.6 km for LTE. HO frequency varies across architectures and bands: NSA deployments experience more frequent HOs than SA deployments due to separate LTE and NR mobility procedures, while mmWave deployments exhibit particularly high HO frequency due to small cell coverage.

In terms of duration, the average HO in NSA 5G takes 167 ms to complete, approximately 1.19× longer than LTE. By decomposing the HO procedure into stages, we find that the *HO preparation* stage—during which base stations evaluate measurement reports and allocate target resources—accounts for 41% of total HO duration in NSA 5G, representing a 48% increase compared to LTE. This longer preparation stage contributes directly to increased data-plane interruption time. We also observe that SA deployments can exhibit comparable preparation delays, reflecting the early-stage maturity of SA network implementations.

We further quantify the energy overhead of mobility. A smartphone traveling at 130 km/h for one hour (without active data transmission) can experience approximately 553 5G HOs, consuming 34.7 mAh of battery capacity, compared to 3.4 mAh for LTE HOs. This highlights the non-trivial energy cost of frequent mobility procedures and underscores the importance of reducing HO-related signaling overhead.

What are 5G HOs' Implications on Carriers? (§3.5) Our analysis also provides insights relevant to network operators. First, we characterize the coverage footprint of 5G cells and its relationship to HO behavior. In NSA deployments, average cell coverage diameters are approximately 1.4 km (low-band), 0.73 km (mid-band), and 0.15 km (mmWave). Because NSA deployments often anchor control-plane signaling on mid-band LTE while data-plane traffic operates on low-band NR, the effective coverage of low-band NR can be constrained by LTE anchor coverage, leading to additional mobility events.

Second, although handovers are intended to improve radio conditions, we observe cases where a 5G-to-5G transition results in reduced throughput, with a median bandwidth reduction of 14% following HO. This occurs because NSA architectures do not support direct gNB-to-gNB handovers; instead, transitions proceed through LTE anchors, causing independent LTE and NR HOs that may not jointly optimize end-to-end performance. Third, we observe that HOs involving co-located LTE and NR cells complete faster than those requiring inter-site coordination, indicating that physical deployment topology significantly affects HO latency.

Can We Predict 5G HOs to Improve Application QoE? (§3.6) Finally, we investigate whether mobility events can be anticipated and leveraged to mitigate application-level disruption. Motivated by the measurement-driven nature of the handover procedure, we design Prognos, a predictive mobility framework that models handover behavior as a closed-loop control process. Prognos leverages cross-layer observations—including signal strength measurements, UE measurement reports, and historical mobility events—to forecast future handovers and their types. The framework adopts a two-stage design: it first predicts short-term signal evolution that determines measurement reports, and then learns the network-side decision logic that maps these reports to handover commands. By decoupling signal prediction from decision modeling,

Prognos reduces model complexity and enables more accurate anticipation of mobility events compared to monolithic prediction approaches.

Evaluation using our dataset shows that Prognos achieves an F1-score between 0.92 and 0.94 for predicting 4G/5G handovers, outperforming prior HO prediction methods by 1.9×–3.8×. We integrate Prognos into adaptive bitrate modules of two representative applications—16K panoramic video streaming and real-time volumetric video streaming—to proactively adapt transmission behavior prior to mobility events. This predictive, measurement-driven control approach yields substantial QoE improvements, including a 34.6%–58.6% reduction in stall time and a 15.1%–36.2% increase in delivered content quality.

Contributions. This chapter makes three primary contributions: (i) the construction of a large cross-layer, multi-band, multi-carrier dataset characterizing mobility in commercial 5G networks; (ii) a comprehensive empirical analysis of HO behavior and its impact on application performance and energy consumption; and (iii) a predictive mobility framework that anticipates HOs and mitigates their impact while remaining compatible with existing 3GPP procedures.

3.2 Measurement Methodology

5G HO Measurement Tool. We extend 5G Tracker [119] to capture several key pieces of information relevant to mobility management in commercial 5G deployments, including Physical Cell IDs (PCIs), handover events, and radio frequency bands. These measurements are extracted using 5G-specific Android APIs introduced in Android 11 [19]. In particular, we leverage the onDisplayInfoChanged() callback of the Android TelephonyManager to infer the operating band

(e.g., low-band, mid-band, or mmWave). The application also logs auxiliary information such as UE geolocation, radio access technology (LTE vs. 5G), and round-trip latency measurements.

5G UE and Other Measurement Tools. We use two commercial 5G smartphones: Samsung Galaxy S21 Ultra 5G (S21U) and Samsung Galaxy S20 Ultra 5G (S20U), equipped with Qualcomm Snapdragon 888 and 865 chipsets, respectively [153, 152]. A total of four devices (three S21U and one S20U) are used throughout the study. These chipsets represent contemporary 5G hardware capabilities, and we expect the measurement findings to generalize to other modern smartphones with comparable modem implementations. To ensure fair comparison across carriers, multiple devices are placed side-by-side during experiments so that external factors such as location, speed, and propagation conditions remain consistent.

Accessing lower-layer signaling information requires vendor-specific diagnostic interfaces. We therefore use Accuver XCAL [18], a professional cellular measurement tool that accesses Qualcomm diagnostic (*Diag*) logs [137]. XCAL runs on a tethered laptop and collects PHY-layer key performance indicators (e.g., PCI and RSRP/RSRQ values) as well as RRC-layer signaling messages such as measurement configurations, handover commands, and reconfiguration procedures [5]. For energy measurements, we use a Monsoon Power Monitor [115] to supply power to the S20U device. All experiments other than energy profiling are conducted using the S21U devices.

5G and 4G Networks. Our analysis spans three dimensions. *(i) Carriers:* measurements are collected across three major U.S. operators (Verizon, T-Mobile, and AT&T). *(ii) Radio Access Technologies (RAT):* we compare LTE, NSA 5G, and SA 5G deployments; at the time of measurement, Verizon and AT&T operated primarily in NSA mode, while T-Mobile deployed both NSA and SA.

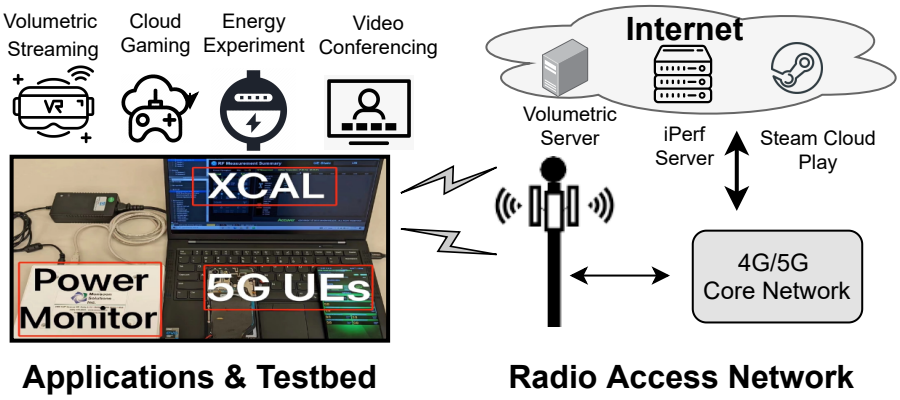


Figure 3.1: An overview of our measurement setup.

(iii) Frequency Bands: the observed bands reflect operator deployment strategies in the measurement regions. For 5G NR, we capture mmWave and low-band data for Verizon and AT&T, and mid-band and low-band deployments for T-Mobile. The LTE dataset includes both low-band and mid-band frequencies across all carriers.

Drive Tests. To study mobility in operational environments, we conduct drive tests across major U.S. cities and interstate highways. Three S21U smartphones—one per carrier—are tethered to a laptop running XCAL via USB 3.0 connections (Figure 3.1). The measurement campaign spans more than 6,200 km of travel. Urban measurements primarily capture dense deployments and mmWave coverage, while highway traces reflect suburban and low-band coverage. This diversity enables analysis of mobility behavior across heterogeneous deployment scenarios. Unless otherwise noted, the majority of data is collected while driving; walking traces are used selectively for specific analyses.

Profiling Applications under Mobility. To evaluate the application-level impact of mobility, we study three representative workloads shown in Figure 3.1: real-time volumetric video streaming, cloud gaming, and live video conferencing. All application experiments are conducted using

Verizon connectivity under LTE, NSA low-band, and NSA mmWave deployments while driving. The detailed setup for each application is described below.

***(i)* Real-time volumetric video streaming.** We use the state-of-the-art ViVo system [70] for volumetric video streaming experiments. A university-hosted server (Intel Xeon, 8 vCPUs, 64 GB RAM, Ubuntu 18.04) with 1 Gbps+ network bandwidth serves the volumetric content. The video is encoded at 30 FPS across five bitrate levels ranging from 43–170 Mbps. The ViVo client runs on an Android S21U smartphone tethered to the XCAL laptop during drive tests. While driving, we replay user viewport traces collected by ViVo and modify the client to log per-frame QoE metrics directly on the device.

***(ii)* Cloud gaming.** To study latency-sensitive interactive workloads, we evaluate three popular cloud-based games: *Brawlhalla*, *CSGO*, and *Hitman 2* [38, 47, 77]. These games are hosted on an AWS EC2 instance (g4dn.2xlarge, 8 vCPUs, 32 GB RAM, NVIDIA T4 GPU, Windows 10, 25 Gbps network) and streamed using Steam Remote Play [160]. Gameplay is accessed through the Steam Link application [159] configured for 4K@60 FPS streaming. Although the S21U device supports up to 2K resolution, 4K frames are transmitted from the cloud and downscaled during rendering. Performance statistics, including latency and frame drops, are collected using Steam's built-in logging tools.

***(iii)* Live video conferencing.** For video conferencing experiments, we use the Zoom application [194] on a smartphone tethered to the XCAL laptop. Following the methodology used in prior studies [109], we conduct a one-on-one video call between a stationary laptop and the mobile UE. During the experiment, we collect Zoom-reported video latency and packet loss statistics to quantify QoE degradation during mobility events.

UDP/TCP Experiments. We further evaluate transport-layer performance under mobility using the iPerf3 bulk transfer tool [80]. Experiments are conducted using both CUBIC [166] and BBR [165] congestion control algorithms. The iPerf server runs on an AWS EC2 instance (g4dn.2xlarge, 8 vCPUs, 32 GB RAM) provisioned with 3 Gbps+ network bandwidth. The server records iPerf logs, packet traces (*pcap*), and socket statistics (*ss*) [99], while the UE runs a cross-compiled iPerf client integrated into the 5G Tracker application.

3.3 Impact of Mobility on Application Performance

In this section, we quantify how mobility events affect representative latency-sensitive and bandwidth-hungry applications. We focus on NSA deployments for application experiments because, at the time of measurement, SA 5G deployments did not consistently deliver the sustained downlink throughput required by our workloads [122].

3.3.1 Quantifying Application QoE under Mobility

We study three applications under mobility: live video conferencing, real-time cloud gaming, and real-time volumetric video streaming.

Live video conferencing. We run a one-on-one Zoom call while driving in an urban loop under NSA 5G coverage. Figure 3.2 shows a representative trace aligned to HO timestamps (green arrows). Around handover events the measured end-to-end video latency increases substantially: on average latency during HOs is 2.26× higher than during non-HO periods (worst-case up to

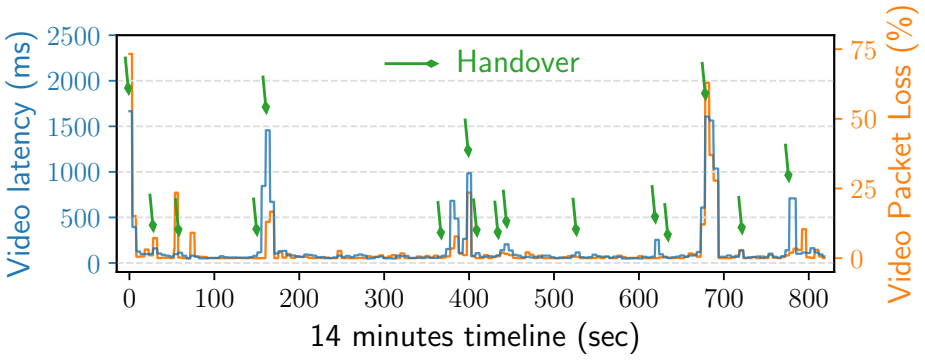


Figure 3.2: Video conferencing latency and packet loss during HOs in NSA 5G (Low-Band).

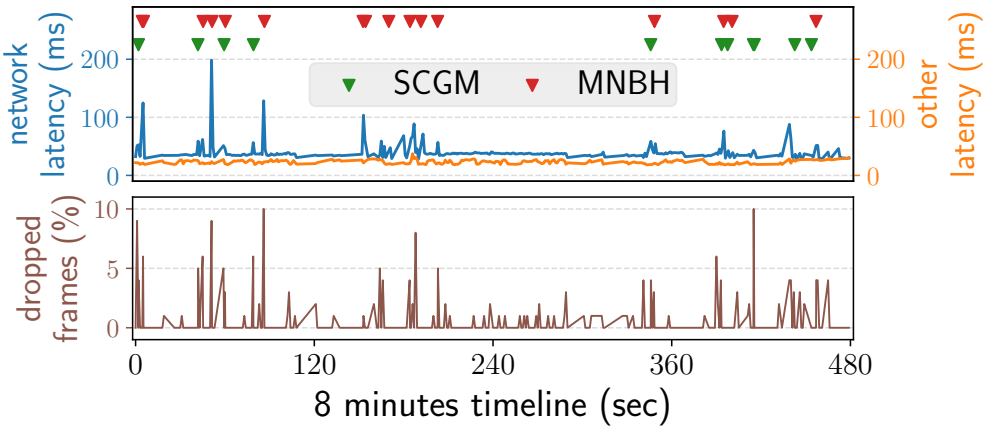


Figure 3.3: Cloud gaming latency and frame drop rate during HOs in NSA 5G.

14.5×). Packet loss reported by the application also increases (average 2.24×). These degradations occur despite available bandwidth that exceeds Zoom's nominal requirement for one-on-one calls [109, 40]. The results highlight that transient disruptions associated with HOs—not only raw link capacity—drive QoE loss for interactive video.

Real-time cloud gaming. We evaluate three cloud-hosted games streamed via Steam Remote Play at 4K@60FPS (frames are downscaled to device resolution). We monitor network transmission latency and dropped frames. During HOs the network latency increases by an average 2.26× (up to 14.5×), and the dropped-frame rate increases by 2.6× for a 60 FPS stream (Figure 3.3).

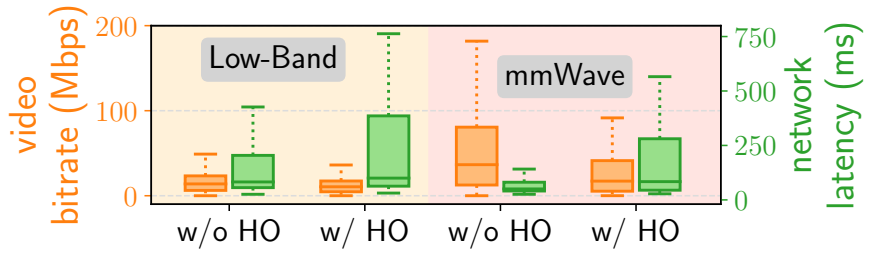


Figure 3.4: Impact of HOs and radio band on the QoE of volumteric video streaming.

Notably, the QoE impact depends on HO type. In NSA deployments, SCG Modification (SCGM, an intra-gNB NR-to-NR change) generally has a smaller QoE penalty than MeNB HO (MNBH), which changes the LTE anchor (Table 2.2). Across our traces, MNBH exhibits on average 16.8 ms higher network latency and 65% more dropped frames than SCGM (see Fig. 3.3). This difference arises because MNBH involves the LTE control-plane anchor and therefore triggers additional cross-technology coordination and reconfiguration that more severely perturbs the user-plane.

Volumetric video streaming. We use ViVo to evaluate volumetric streaming performance across two NR bands (low-band and mmWave). We measure per-frame bitrate and network latency to capture the combined effects of throughput and responsiveness. Figure 3.4 compares QoE metrics across bands: median video bitrate drops by 31% for low-band HOs and by 58% for mmWave HOs. Median network latency increases by 41% for low-band and by 107% for mmWave. MmWave HOs therefore produce substantially larger and more volatile QoE degradation—sometimes producing multi-Gbps throughput drops—whereas low-band HOs produce smaller but still noticeable degradation. Overall, volumetric results reinforce that QoE fluctuation depends jointly on HO type, radio access technology, and frequency band.

3.3.2 *5G-only vs. dual* traffic mode in NSA Deployments

NSA supports multiple traffic modes for the user plane: an MCG-split *dual mode* that splits data across LTE and NR, and an *SCG* or *5G-only* mode that carries user data solely on NR [6]. These modes affect how handovers perturb transport-layer performance.

Figure 3.5 shows TCP (BBR) RTT measured during HOs for traces collected under the two modes. We draw three conclusions. First, in the absence of handovers (*w/o HO*), 5G-only mode achieves lower RTT than dual mode on median, consistent with a shorter data-path when traffic is delivered directly to the gNB. Second, during HOs, dual mode exhibits smaller median RTT inflation (1–4%) because the LTE interface can continue carrying user traffic while NR experiences reconfiguration; the presence of the LTE path therefore absorbs some HO-induced fluctuations. Third, in 5G-only mode, HOs cause larger RTT inflation (median increases of 37–58%), since no secondary interface is available to maintain steady traffic. We observe similar behavior for TCP Cubic.

These observations illustrate a trade-off: dual mode offers resilience to HO perturbations at the cost of somewhat higher baseline RTT (due to split/data-forwarding paths), while 5G-only mode reduces baseline RTT but is more sensitive to HO disruptions. Operators can exploit this trade-off (for example, by using split-mode with direct core-to-gNB forwarding) to balance performance and robustness for latency-sensitive services.

Summary. Across applications and transport tests, handovers produce measurable QoE degradation. The severity of impact depends on HO type (SCGM vs MNBH vs SCGR, etc.), frequency band (mmWave $\gg$ mid/low), and NSA traffic mode (dual vs 5G-only). These results motivate

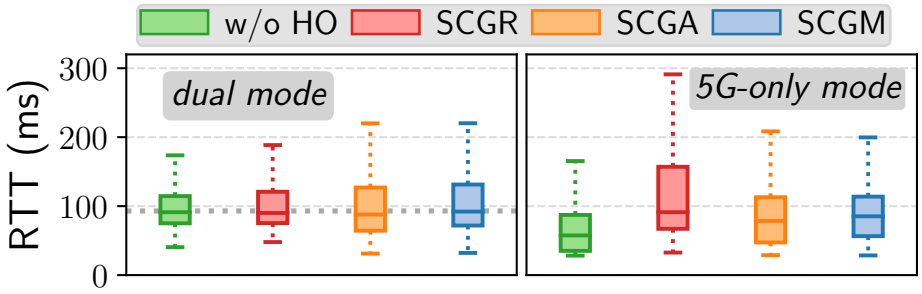


Figure 3.5: TCP (BBR) RTT during HOs in two NSA deployment modes.

predictive, measurement-driven approaches that anticipate mobility events and enable proactive application or network-side adaptation.

3.4 Characteristics of 5G Handovers

Motivated by the application-level findings in §3.3, we now present a systematic characterization of handover (HO) behavior in commercial 5G networks. Using our large cross-country dataset, we focus on three dimensions that directly affect UE performance: *HO frequency*, *HO duration*, and *HO energy consumption*.

3.4.1 Handover Frequency

We quantify HO frequency using our drive-test traces, comparing across radio access technologies (4G vs 5G), deployment architectures (SA vs NSA), and frequency bands (low, mid, mmWave). Our freeway traces (Table 3.1) show that HOs are considerably more frequent in many 5G scenarios than in LTE. In particular, NSA 5G HOs occur every 0.4 km on average, compared to every 0.6 km for LTE. SA 5G shows lower HO frequency in our traces (0.9 km), suggesting potential reductions in HO overhead in mature SA deployments [143].

HO frequency also varies substantially by band. Within NSA deployments we observe HO intervals of roughly 0.13 km for mmWave, 0.35 km for mid-band, and 0.4 km for low-band. The extreme frequency at mmWave reflects the much smaller coverage footprints and the need for frequent beam/cell transitions (see §3.5.1).

We also measure HO-related signaling overhead across layers. Focusing on RRC messages (Measurement Report, RRC Reconfiguration, RRC Reconfiguration Complete [5]), MAC-layer Random Access (RACH) procedures [8], and PHY-layer SSR/beam procedures, we find that SA 5G reduces total HO-related signaling by approximately 3.8× relative to LTE in our traces (largely due to lower HO frequency). Conversely, PHY-layer signaling (including beam management) increases by more than 5× in NSA mmWave relative to low-band NSA, driven by frequent beam operations and small cell coverage [13, 121].

3.4.2 Handover Duration

Long HO durations are a primary driver of application degradation (cf. §3.3.1). We decompose total HO time into two phases consistent with RRC procedures: the *preparation* stage (T_1), during which the network evaluates measurement reports and reserves resources at the target; and the *execution* stage (T_2), during which the UE completes the RRC reconfiguration and RACH to attach to the target cell.

Overall, NSA 5G HOs are longer on average than LTE HOs. The mean HO duration for NSA 5G in our traces is 167 ms, compared to 76 ms for LTE; SA 5G shows an intermediate mean ($\sim$110 ms) with higher variance.

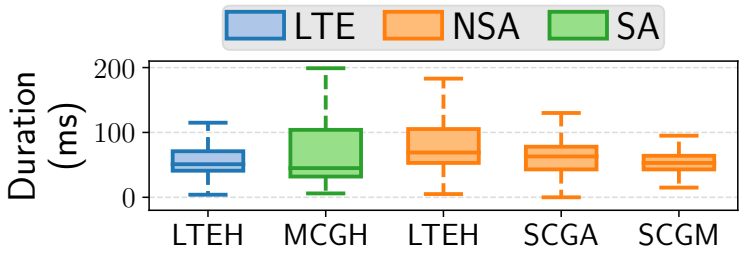


Figure 3.6: HO preparation stage (T_1) for T-Mobile in NSA 5G *vs.* SA 5G *vs.* LTE.

HO preparation (T_1). The preparation stage is responsible for a large fraction of the total HO delay in NSA. In our dataset T_1 accounts for ∼41% of the total HO duration in NSA 5G. Figure 3.6 presents per-carrier measurements of T_1 for LTE, NSA, and SA. For example, carrier T-Mobile exhibits an average T_1 that is 92 ms (48%) longer in NSA than in LTE. We attribute this increase to the additional cross-node signaling and coordination required in NSA architectures (eNB↔gNB), particularly when the nodes are not co-located [6, 142]. SA preparation times are comparable to LTE on median but show larger variability in our traces, likely reflecting early-stage SA deployments.

HO execution (T_2). Execution time directly impacts user-plane interruption and thus application QoE. In NSA 5G T_2 comprises 59% of total HO duration. Figure 3.7 compares T_2 across technologies and bands: mmWave incurs 42–45% larger execution delays than low-band within NSA even though PRACH formats in mmWave can be shorter [8]. We believe the increased T_2 in mmWave stems from beam-management procedures (tracking, search, and selection) and other PHY-layer operations that add latency during attachment to the target beam/cell [13, 121]. Overall, NSA-induced signaling and beam operations explain the longer end-to-end HO times and the associated application-level impacts documented earlier.

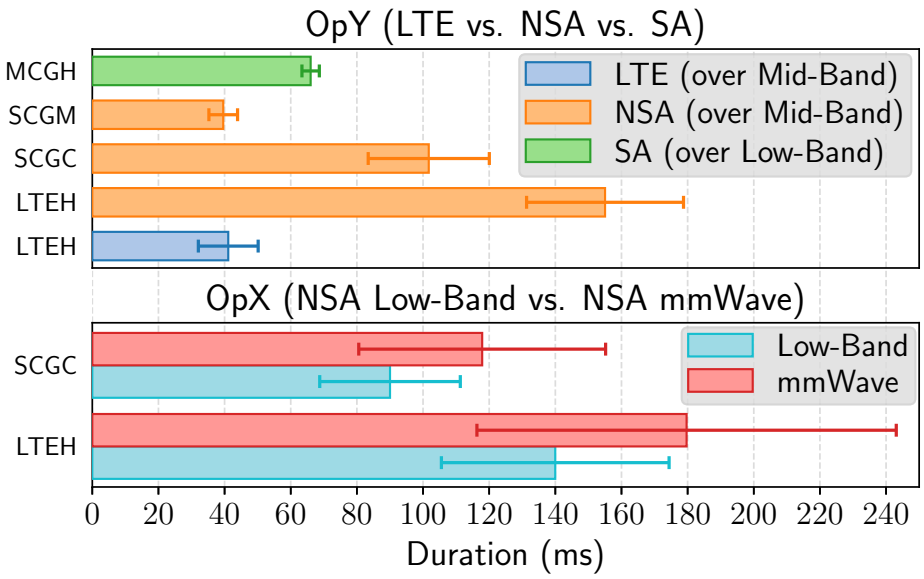


Figure 3.7: Comparison of HO execution stage T_2 across access technologies (NSA 5G *vs.* SA 5G *vs.* LTE) and radio bands (Low-Band *vs.* mmWave).

3.4.3 Handover Energy Consumption

Frequent HOs also impose an energy cost on UEs. We quantify this cost using Monsoon Power Monitor measurements synchronized with XCAL-derived lower-layer HO events. Because XCAL and Monsoon cannot be connected simultaneously to the same device, our measurement procedure is as follows: (i) use XCAL to identify and validate repeatable HO trigger locations over a surveyed route; (ii) verify event correspondence between 5G Tracker (Android APIs) and XCAL; (iii) perform repeated drive loops at identified locations while recording power with Monsoon and HO events with 5G Tracker. During Monsoon runs, we maintain the UE in RRC_CONNECTED by sending a small ping every 5 s and subtract baseline/ping energy from the HO measurements.

Figure 3.8 reports both per-HO power and energy per unit distance (combining per-HO energy with HO frequency from §3.4.1). Key results: a smartphone traveling at 130 km/h for one hour encounters on average 553 NSA low-band HOs, consuming 34.7 mAh; for NSA mmWave the analogous numbers are 998 HOs consuming 81.7 mAh. By comparison, 4G HOs consume 3.4 mAh

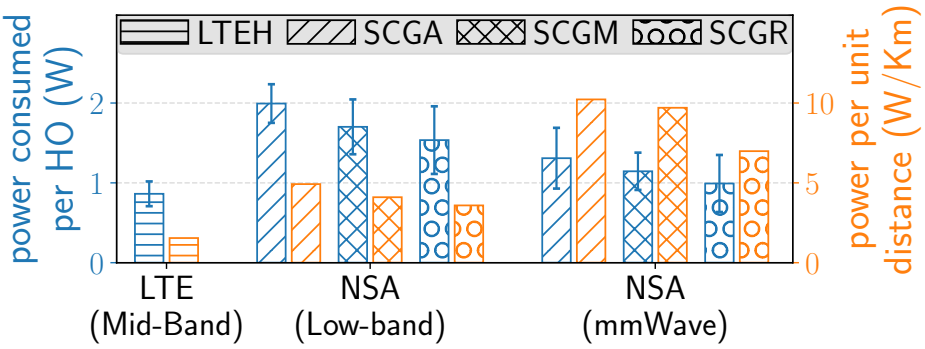


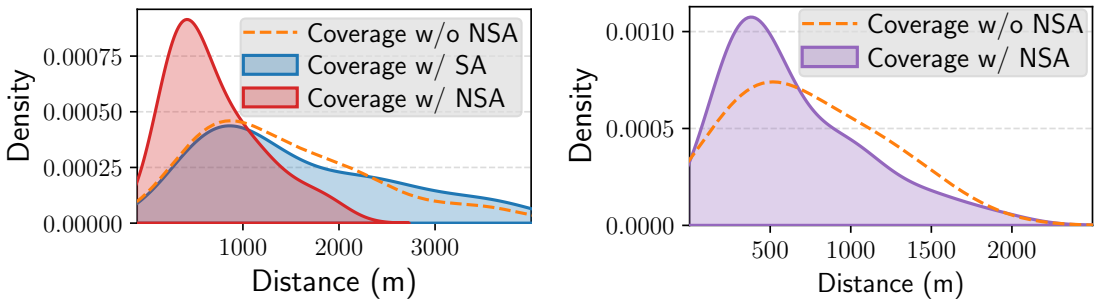
Figure 3.8: Comparing power consumption of HOs in Low-Band NSA 5G *vs.* mmWave NSA 5G *vs.* Mid-Band LTE.

in the same scenario. On a per-HO basis, NSA 5G HOs consume 1.2–2.3× more energy than LTE HOs, since both 4G and 5G radios participate in many NSA procedures. Interestingly, a single mmWave HO can be 54% more energy efficient than a low-band HO—likely due to more efficient PRACH formats in some mmWave configurations [8]—but the much higher HO frequency at mmWave yields substantially larger cumulative energy per unit distance (1.9–2.4× higher than low-band).

These measurements demonstrate that HO energy is non-negligible for mainstream smartphones and becomes especially important for devices with constrained power budgets. The combined cost of frequent HOs (higher signaling, longer preparation/execution delays, and higher cumulative energy) motivates measurement-driven, predictive interventions that reduce unnecessary mobility events or enable graceful mitigation when HOs are imminent.

3.5 Implications of 5G Handovers on Carriers

This section examines handovers (HOs) from the network operator's perspective. Building on the empirical characterizations above, we (i) map a coverage landscape for 5G and identify an



(a) Low-Band

(b) Mid-Band

Figure 3.9: Comparison of tower's effective coverage footprint (diameter): with and without NSA.

NSA-related coverage effect, (ii) quantify how HOs affect observed bandwidth, and (iii) expose operational challenges caused by non-co-located LTE (eNB) and NR (gNB) equipment.

3.5.1 Coverage Landscape in 5G

Figure 3.9 summarizes per-cell effective coverage estimated from our drive traces (we approximate a cell's diameter as the continuous ground distance during which the UE remains attached to the same PCI). Across NSA traces, the estimated average cell diameters are approximately 1.4 km (low-band), 0.73 km (mid-band), and 0.15 km (mmWave). Low→mid-band coverage drops by roughly 48%, and mmWave coverage is an order of magnitude smaller than low-band, reflecting the well-known frequency-dependent attenuation and deployment densities.

NSA reduces effective low-band coverage. A notable finding is that the *effective* coverage of a low-band NR cell in NSA deployments is substantially smaller than the same band in SA. In our traces the effective low-band coverage under NSA is reduced by roughly 1.2–2× compared to SA (see Figure 3.9(a)). The root cause is architectural: in NSA the NR data plane (low-band) is often anchored to an LTE control plane operating on a different band (commonly mid-band). Because NSA control-plane mobility (the LTE anchor) can trigger NR HOs, an NSA control-plane change

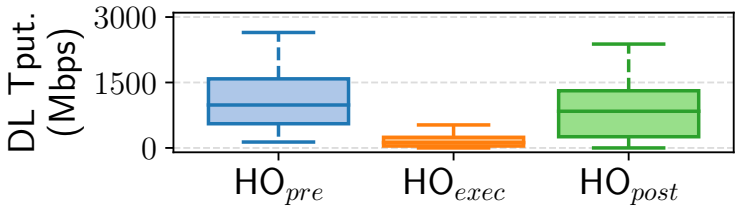


Figure 3.10: Impact of SCGC on network bandwidth in 5G mmWave.

will often induce additional NR HOs even when the NR link alone would not have required them. In effect, NSA coupling can negate low-band NR's larger propagation footprint and increase HO frequency compared to an equivalent SA deployment.

3.5.2 Impact of 5G HOs on Bandwidth

Horizontal HOs are expected to improve end-user throughput by connecting a UE to a stronger serving cell. However, operator traces in 5G reveal counterexamples where inter-gNB HOs (SCG Change / SCGC) yield reduced throughput after the HO. In an mmWave walking experiment with continuous bulk downloads, we measure throughput immediately before (HO$_{pre}$), during (HO$_{exec}$), and just after (HO$_{post}$) the HO. Figure 3.10 shows that median post-HO throughput is 14% lower than pre-HO throughput for SCGC events.

We attribute this unexpected regression to NSA's HO sequencing. Because NSA does not support a direct gNB→gNB HO, an apparent 5G→5G transition proceeds as a 5G→4G followed by a 4G→5G sequence, with each leg decided and executed independently. The intermediate LTE-anchored step (and independent per-node decision logic) can prevent the network from selecting a final target that improves aggregate 5G performance. In short, per-node, threshold-based decisions made without an end-to-end view of the composite 5G→5G transition can produce suboptimal outcomes for throughput.

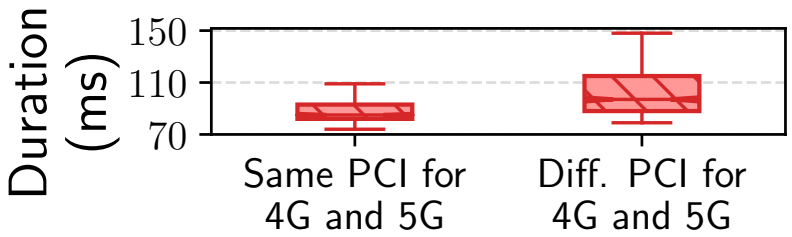


Figure 3.11: Handover Duration (T_1 + T_2) with same (vs. different) 4G-LTE PCI and 5G-NR PCI.

3.5.3 Impact of eNB and gNB Co-location

In NSA the UE maintains connections to both an eNB (LTE) and a gNB (NR); these logical nodes may or may not be deployed at the same physical tower. We use PCI geography as a heuristic for co-location: when the 4G and 5G PCIs observed in the same geographic neighborhood produce overlapping convex hulls, we treat the nodes as *co-located*; non-overlapping hulls indicate distinct physical sites.

Across our dataset, only about 5%–36% of NSA low-band samples show co-located eNB and gNB pairs (variation depends on carrier and region). Crucially, HO latency is sensitive to co-location: HOs where the eNB and gNB are co-located complete faster than those requiring inter-site coordination. Figure 3.11 shows that same-PCI (co-located) NSA HOs save on average 13 ms compared to different-PCI (non-co-located) HOs. The extra delay when nodes are separated is consistent with additional signaling and cross-tower coordination overheads documented in vendor interworking reports.

Operator implications. Taken together, these findings have immediate implications:

• **Reconsider NSA anchoring policies.** NSA's LTE-anchored control plane can shrink effective NR coverage and increase HO frequency; where feasible, operators should migrate coverage and control to SA topology or reduce cross-plane HO coupling for low-band NR.

• **Adopt sequence-aware HO logic.** Inter-gNB transitions that are implemented as separate per-node decisions can produce net throughput regressions. Operators can improve post-HO performance by evaluating the end-to-end consequence of composite HO sequences (5G→4G→5G) before committing resources.

• **Co-locate LTE/NR where practical.** Co-location reduces HO latency and the need for cross-site coordination. When co-location is infeasible, HO decision logic should account for physical tower topology (PCI geography) to avoid costly cross-site handovers.

In the next section, we exploit these operational insights to design predictive features and mitigation strategies that anticipate problematic HOs and reduce their application-level impact.

3.6 4G/5G Handover Prediction

In this section, we formalize the HO prediction problem and present Prognos, our measurement-driven prediction system. Prognos is designed to be lightweight, explainable, and adaptive to carrier-specific HO logic; it produces both HO type predictions and a compact utility value (`ho_score`) that applications can use to adjust behavior proactively. We evaluate Prognos against prior approaches and show its benefit when integrated into two demanding applications: 16K panoramic VoD and real-time volumetric streaming.

3.6.1 Challenges and Goals

Predicting HOs in operational cellular networks is difficult for three related reasons. First, carrier HO logic is essentially a "black-box": operators use vendor- and region-specific policies and

thresholds that differ across geography and over time. Second, HOs are the result of a short decision horizon at the network: once a measurement report (MR) is triggered and delivered, only a few tens of milliseconds remain before the network may issue an HO command. Third, datasets are imbalanced: HOs are rare relative to continuous signal samples, which complicates purely data-driven, monolithic learning.

From these observations, we derive our system goals. Prognos must be:

- **Explainable**: allow understanding of the carrier-specific logic inferred from data and enable sanity checks during prediction.
- **Adaptive and transferable**: incrementally learn and update HO patterns so the model generalizes across geographic regions and carriers.
- **Light-weight and reactive**: run on resource-constrained UEs with low latency so applications have lead time to react.
- **Context-aware**: incorporate RAT (LTE/NR), band, and other context to produce actionable outputs (HO type + `ho_score` representing expected capacity change).

To meet these goals, we adopt a decomposed, incremental architecture that mirrors the cellular control loop: predict the measurements (what the UE will report), learn the network decision logic (how the network maps reported measurements to HO commands), and then produce HO forecasts that applications can act upon.

3.6.2 System Design

Figure 3.12 shows Prognos's high-level architecture. The system has three core components. The *report predictor* module considers mobility configurations and signal strength qualities to predict

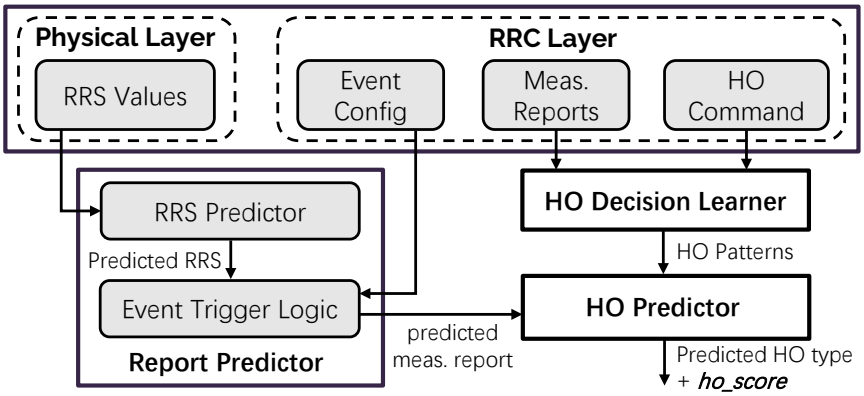


Figure 3.12: Design of HO prediction system Prognos.

MRs. The *decision learner* module learns the carrier-specific HO decision logic by leveraging ideas from sequential pattern mining. Finally, the *handover predictor* module uses the sequence of predicted MRs and learned HO logic to forecast the HO type. The three modules operate in a streaming, online manner and collectively produce the final prediction and `ho_score`.

Measurement report prediction. Predicting the MR before it is actually reported gives applications valuable lead time. To forecast whether an MR will be triggered in the next prediction window, we combine three inputs: (i) the serving-cell measurement configuration (thresholds, time-to-trigger, TTT), (ii) a short-term forecast of the serving-cell reference signal strength (RRS), and (iii) a short-term forecast of the neighbor-cell RRS. For signal forecasting, we use a light-weight linear regression over a recent history window; before regression we apply a triangular-kernel smoothing to reduce small-scale fading and measurement noise. Given the predicted RRS values and the configured trigger conditions (Table 2.1), the module reports whether an MR event will be raised and, if so, when (respecting the configured TTT). In our traces, *report predictor* yields an average lead-time improvement of 931 ms over waiting for the MR (Fig. 3.13) with a small loss in accuracy.

Policy-based HO decision learning. Carriers implement HO decisions as policy logic that maps MR sequences to HO commands. We model this logic as repeated *patterns*: sequences of MRs that historically precede a particular HO type. Input to *decision learner* is the streaming sequence of observed MRs and HO commands (from RRC traces or inferred via Android APIs + XCAL). We split the stream into *phases* where each phase contains one or more MRs followed by the HO command (if any). The learner extracts frequent sequential patterns using an online adaptation of prefixSpan [134], incrementing support counts when old patterns reappear and adding fresh patterns when novel sequences are observed. We age and evict patterns based on freshness to bound memory and to track evolving operator logic. Each learned pattern is thus associated with its HO type, support, length, and last-observed timestamp (freshness).

Handover prediction. At prediction time we match the sequence of predicted (or observed) MRs in the current phase against the catalog of learned patterns produced by *decision learner*. If no pattern matches, *handover predictor* emits "no HO." Otherwise, it selects the pattern with the highest similarity score; similarity is a weighted function of pattern support, length, and freshness. The predicted HO type is returned together with a confidence value. We convert the predicted HO type and historical throughput deltas associated with that HO type into a compact `ho_score` in $(0, \infty)$ that applications can use to scale throughput forecasts and adjust behavior. Concretely, `ho_score` is computed as the median ratio of post-HO to pre-HO throughput observed historically for the predicted HO type (so a score of 0.4 indicates an expected 60% throughput drop).

Table 3.2: Prognos performance evaluation on D1 and D2.

Dataset	Method	F1-Score	Precision	Recall	Accuracy
Dataset D1					
D1	GBC	0.475	0.403	0.577	0.936
D1	Stacked LSTM	0.284	0.190	0.562	0.857
D1	Prognos (ours)	**0.919**	**0.928**	**0.917**	**0.917**
Dataset D2					
D2	GBC	0.396	0.346	0.463	0.867
D2	Stacked LSTM	0.241	0.144	0.732	0.420
D2	Prognos (ours)	**0.936**	**0.946**	**0.926**	**0.931**

3.6.3 Performance Evaluation

We evaluate Prognos via trace-driven emulation using collected traces and compare against two prior approaches: (i) a Gradient Boosting Classifier (GBC) similar to Mei et al. [112] that uses lower-layer features, and (ii) a stacked LSTM approach that uses location/time series [130]. All approaches use a 1 second history and 1 second prediction window unless otherwise noted. For the GBC and LSTM baselines, we perform standard offline training (60% train / 40% test); Prognos learns online from the stream and does not require a separate offline training phase.

Datasets. We use two datasets collected for Verizon (VZW) at 20 Hz: D1 — seven traces of a 35 min walking loop with mmWave and LTE mid-band — and D2 — ten repeats of a 25 min downtown loop that also contains low-band NR. D1 contains over 320 HOs; D2 contains over 840 HOs. HO events are rare (∼0.4% of samples), so we report F1, precision, and recall in addition to accuracy.

Results. Table 3.2 summarizes prediction performance. Prognos achieves F1 scores of 0.919 and 0.936 on D1 and D2 respectively, far outperforming GBC and stacked LSTM baselines on F1 and precision. The baselines sometimes reach high raw accuracy (dominated by the majority

"no-HO" class) but suffer low F1 because they miss or misclassify the rare HO events. Prognos 's decomposition (MR forecasting + policy learning) yields robust, high-quality predictions without requiring large offline models.

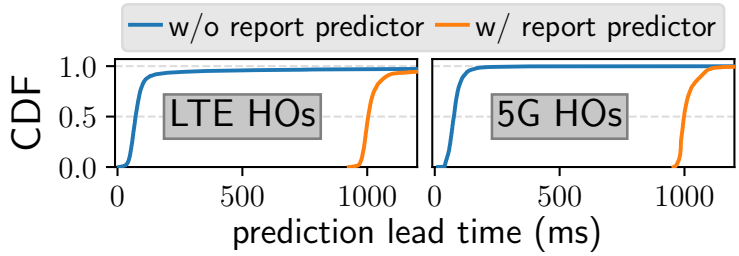


Figure 3.13: Lead-time improvement in HO prediction enabled by *report predictor*.

The *report predictor* component contributes substantial lead time (median +931 ms) which is critical for application-level mitigation. Figure 3.13 quantifies the lead-time improvement achieved by *report predictor* for LTE and NR HO prediction. By forecasting MRs prior to their transmission and learning base-station decision logic, Prognos predicts HO events earlier than reactive baselines.

Because Prognos learns HO patterns online, prediction accuracy is initially limited by the availability of observed transitions. Across datasets D1 and D2, the F1-Score exceeds 0.9 after approximately 14 and 11 minutes, respectively, reflecting the time required to accumulate representative HO samples. Convergence time scales with HO frequency, with faster mobility reducing the required observation window. Figure 3.14 evaluates a simple mitigation strategy. Bootstrapping Prognos with the most frequent HO pattern extracted from historical traces increases the F1-Score to $\sim$0.8 within 1.5 minutes, compared to substantially lower accuracy without bootstrapping. Alternatively, the system can defer predictions during the initial phase and operate in observation-only mode until sufficient data is collected.

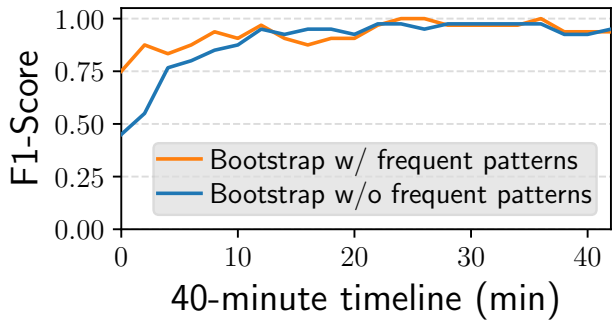


Figure 3.14: Impact of bootstrapping with most frequent pattern during startup phase of Prognos.

3.6.4 Prognos Use Cases

We demonstrate the application value of Prognos by integrating its predictions into two resource-demanding video workloads. For both, we use trace-driven emulation (Mahimahi [123]) and a set of 40+ representative bandwidth traces (each 240 s) collected by saturating a downlink while driving in VZW coverage (NR low-band, mmWave; LTE mid-band). We only retain traces with average bandwidth below 400 Mbps and minimum above 2 Mbps to avoid trivial cases.

16K panoramic VoD. We extend a Pensieve-style ABR evaluation with a custom 16K panoramic video encoded at six quality levels (720p, 1080p, 2K, 4K, 8K, 16K) split into 60 chunks (120 s total). For each ABR algorithm (rate-based, fastMPC, robustMPC) we scale the throughput prediction at chunk decision time by `ho_score` when Prognos predicts an imminent HO; otherwise we leave the prediction unchanged. Compared to the original ABR, Prognos-enhanced schemes reduce stall time by 34.6–58.6% and improve average delivered quality; throughput prediction error during HOs is reduced by 52.4–61.3% (Figure 3.15a–b). In absolute terms the HO-aware algorithms approach the ground truth performance within small margins.

Real-time volumetric streaming. We emulate ViVo and FESTIVE-style adaptation over the same traces. The volumetric content is compressed into 3 min segments at five point-cloud

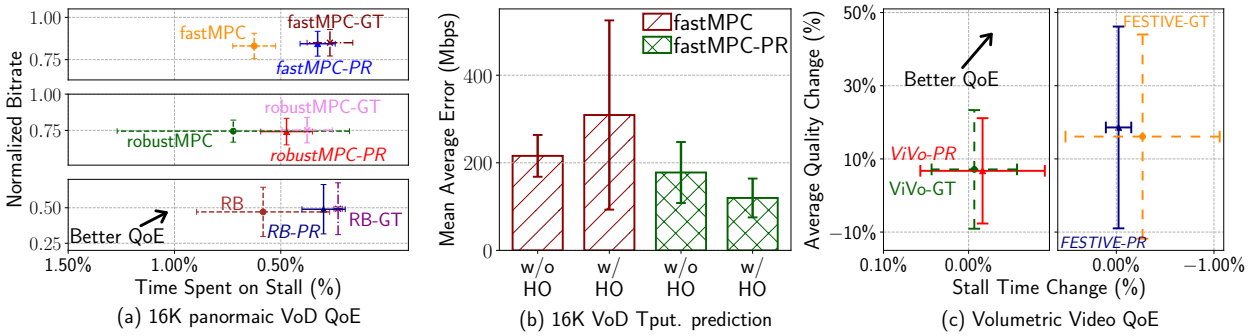


Figure 3.15: QoE improvement due to Prognos for *16K panoramic VoD* and *real-time volumetric video streaming*.

density levels (bitrates 43, 77, 110, 140, 170 Mbps). By applying `ho_score`-adjusted throughput predictions, Prognos-aware adaptations increase delivered quality by 15.1–36.2% while also reducing stall time by 0.24–3.67% relative to the original algorithms (Figure 3.15c). The HO-aware algorithms approach the ground-truth HO-informed upper bound closely (within small absolute margins).

Implementation notes. Prognos learns new HO patterns at a modest rate (9.1 $\pm$ 2.3 patterns per hour) and evicts old patterns at a comparable rate (8.3 $\pm$ 3.1 per hour) to bound memory and adapt to policy changes. The runtime components (linear regression + pattern matching) are intentionally light-weight so they can run on-device with low energy cost.

3.7 Summary and Broader Implications

Mobile systems, particularly 5G, operate at the intersection of multiple interacting variables, many of which are controlled by cellular carriers and evolve over time. In this work, we presented a comprehensive measurement-driven study of 5G mobility management, supported by over 6,200 km of drive tests spanning diverse geographic and temporal conditions. Our analysis revealed that HOs in modern 5G deployments are both more frequent and more disruptive

than in prior generations, leading to measurable impacts on application QoE, transport-layer performance, and UE energy consumption. Motivated by these findings, we designed Prognos, a predictive HO framework that anticipates mobility events using cross-layer signals and enables proactive mitigation of application-level disruptions.

Despite the breadth of our study, several limitations remain. First, our measurements were conducted without direct collaboration with cellular carriers, limiting visibility into vendor-specific implementations, proprietary HO policies, and base-station scheduling behavior. Second, while our energy analysis complements prior work on device-specific power characteristics [122, 182], detailed data-plane energy modeling across device models and chipset generations remains outside our scope. Third, although factors such as mobility speed, tower density, and user load are known to influence performance [65, 17, 177, 50], 5G mobility introduces additional complexity due to heterogeneous bands, NSA architectures, and dense deployments. By conducting measurements across multiple locations, times of day, and weeks, we mitigate—but do not fully eliminate—the impact of such confounding factors.

Our findings remain relevant as 5G continues to evolve. Current deployments are largely NSA-based, with LTE providing control-plane support; however, future transitions toward SA architectures and new NSA configurations will retain similar mobility dynamics. Multiple 3GPP deployment options allow operators to gradually migrate toward fully 5G-native control and data planes, and our HO prediction framework is compatible with both LTE and 5G HO procedures. As cellular systems transition toward 5G-Advanced and eventually 6G, understanding the operational implications of these architectural shifts will remain essential, and measurement-driven studies such as ours provide a foundation for that analysis.

Finally, our work highlights the growing importance of cross-layer observability in mobile systems. Accurate HO prediction and mitigation require access to measurements across PHY, MAC, RRC, and transport layers—information that is currently accessible only through specialized tools such as XCAL, MobileInsight, and USRP-based decoders. Emerging 5G MEC frameworks and Radio Network Information (RNI) APIs offer a path toward exposing such signals to applications and edge services. We argue that enabling controlled access to lower-layer information through standardized interfaces would unlock new opportunities for throughput prediction, latency optimization, energy modeling, and mobility-aware application adaptation. By releasing our datasets and analysis artifacts, we hope to facilitate future research that advances cross-layer, measurement-driven design in next-generation mobile networks.

Chapter 4

Dynamic Duplexing Under Asymmetric 5G Workloads

4.1 Introduction

5G NR introduces substantial flexibility in radio resource allocation, enabling higher data rates and lower latency for emerging applications such as augmented reality (AR) [31, 27], autonomous driving [97], remote healthcare [173], and real-time video analytics [24, 100]. To support these workloads, more than 80% of 5G operators deploy Time Division Duplex (TDD) [76], whereas earlier generations (e.g., LTE and 3G) relied primarily on Frequency Division Duplex (FDD) [60]. TDD enables uplink (UL) and downlink (DL) transmissions to share the same spectrum through time-slot scheduling, providing the potential for dynamic UL/DL adaptation based on traffic demand.

5G NR further introduces *dynamic* TDD, allowing the base station (BS) to adjust UL/DL slot allocation through configurable TDD policies [3]. Although 3GPP specifies signaling mechanisms for dynamic TDD, the design of practical TDD policies is left to operators. Consequently, current deployments often rely on static or reactive policies that do not fully exploit the flexibility of dynamic TDD. Prior work largely evaluates dynamic TDD using analytical models or simulation

studies [154, 107, 149, 29], and therefore provides limited insight into real-world application-level impact or deployment constraints.

To bridge this gap, we conduct a measurement-driven study of TDD configurations in operational 5G networks and quantify their impact on application QoE (§4.2). Our measurements reveal three key findings. First, BS traffic demand fluctuates rapidly over short timescales, rendering static TDD policies ineffective for latency-sensitive and UL-intensive workloads. Second, reactive policies that adjust UL/DL ratios based solely on past demand incur performance loss compared to proactive adaptation. Third, beyond slot ratios, the temporal arrangement of UL and DL slots significantly affects latency, throughput stability, and application QoE. These observations motivate a predictive, cross-layer approach to TDD policy design.

The goal of this work is to design a practical TDD policy adaptation system that dynamically adjusts both UL/DL slot distribution and slot arrangement to improve application QoE, without requiring explicit QoE feedback from UEs or applications. This problem presents several challenges. *First*, the TDD policy space is combinatorial, with numerous feasible slot arrangements. *Second*, traffic load and channel conditions fluctuate rapidly, requiring timely adaptation. *Third*, BSs lack direct knowledge of application-level QoE objectives, necessitating indirect optimization through QoS metrics. *Finally*, frequent TDD reconfiguration can disrupt transport-layer congestion control and application rate adaptation, while inherent UL/DL asymmetry further complicates policy optimization.

To address these challenges, we present Wixor, a practical TDD policy adaptation system for 5G/xG radio access networks. Wixor decomposes the problem into two components. First, it predicts short-term UL/DL demand using a proactive demand customization engine (§**??**), leveraging BS-level features and learning-based forecasting to handle dynamic traffic and channel variations.

Second, given the predicted slot distribution, Wixor derives the optimal UL/DL slot arrangement through a smart policy provision framework (§**??**), optimizing QoS metrics while applying conservative smoothing to avoid abrupt policy changes. This design enables incremental deployment at the BS, requires no UE or application modifications, and is fully compliant with 3GPP signaling procedures.

We prototype Wixor on a programmable 5G testbed using an open-source cellular stack [158], implementing over 2.3K lines of code. We evaluate Wixor through over-the-air experiments and trace-driven simulations using diverse mobility scenarios (driving, walking, *etc.*) and real applications (edge video analytics, live conferencing, *etc.*). Our key findings are as follows.

- Across six application workloads, Wixor improves QoE metrics by 2.5%–96.5% compared to existing baselines, and performs within 91.6% of an oracle policy (§4.7.2, §4.7.3).
- Wixor adapts effectively to dynamic traffic conditions, achieving up to 3.2% higher throughput and 15.3% lower latency in driving scenarios compared to static policies (§4.7.4).
- Wixor remains lightweight, scalable, and compliant with RAN constraints, operating close to the optimal solution across diverse configurations (§4.7.4, §4.7.5).

4.2 Motivation & Challenges

This section presents measurement-driven evidence highlighting the need for dynamic TDD policy adjustment and the practical challenges in designing such a system.

4.2.1 Experiment Setup

Live 5G experiments. To characterize TDD policies employed by today's 5G operators, we build a live 5G measurement platform. We use NG-Scope [180] to decode control-channel information from a base station, including Transport Block Size (TBS) and Modulation and Coding Scheme (MCS) for active UEs. Simultaneously, a Samsung Galaxy S22+ device connected to the same BS collects lower-layer TDD configuration using Accuver XCAL [178]. Using this setup, we collect over 26 hours of measurements across different times of day from three operators: T-Mobile (Band 41 @ 2500 MHz), Verizon (Band 77 @ 3700 MHz), and AT&T (Band 77 @ 3700 MHz). We refer to this dataset as D3.

Over-the-air 5G testbed. To evaluate controlled scenarios, we build an end-to-end in-lab 5G testbed comprising two Google Pixel 7 devices, a programmable BS, and a 5G Core. The BS consists of an srsRAN-based eNB/gNB stack running on an Intel Core i7 laptop, with a USRP B210 SDR configured for Band 78 @ 3410 MHz. The Open5GS core runs on a separate machine. Experiments are conducted in a 20 m $\times$ 12 m room with controlled mobility, while application servers are hosted locally with configurable network delay. All experiments are automated using ADB scripts and repeated at least five times.

Trace-driven simulator. To enable reproducible experiments and accelerate model training, we build a trace-driven simulator using ns-3 5G-Lena [16]. The simulator configuration mirrors the over-the-air testbed. We replay SINR traces from prior 5G measurement studies [122, 72, 61], resulting in a corpus of 200 traces (each 300 seconds). Randomly sampling traces for different UEs creates heterogeneous channel conditions representative of real deployments.

Table 4.1: Application workloads used in this paper (UL = uplink, DL = downlink; Lat. Sen. = latency-sensitive, Bwd. Int. = bandwidth-intensive).

App name	UL Lat. Sen.	UL Bwd. Int.	DL Lat. Sen.	DL Bwd. Int.
Edge Video Analytics (EVA)	✓	✗	✓	✗
Edge-assisted Vehicle Perception (EVP)	✓	✓	✓	✗
Live Video Ingest (LVI)	✗	✓	✗	✗
Video-on-Demand (VoD)	✗	✗	✗	✓
Live Video Conferencing (LVC)	✓	✗	✓	✗
HTTP File Transfer (HFT)	✗	✓	✗	✓

Background traffic. To emulate realistic traffic at scale, we generate UL/DL background traffic using real-world traces derived from NG-Scope measurements. Each trace represents aggregated UL/DL traffic from multiple UEs. During experiments, one Pixel device generates application traffic while the second device replays background traffic. In simulation, traffic is generated for the number of UEs specified in each trace.

Applications. We develop a suite of latency-sensitive and bandwidth-intensive applications (Table 4.1) to evaluate the impact of TDD policies. These include live video ingest, edge video analytics, and other emerging workloads that exhibit diverse UL/DL demand patterns.

4.2.2 Need for Dynamic TDD Policy Adjustment

Static, DL-biased TDD policies cannot adapt to dynamic traffic load. Using the D3 dataset, we analyze temporal variations in BS traffic load and network response. Our analysis reveals four key insights:

(i) Rapid load fluctuations. Figure 4.1 shows significant short-term variability in traffic load. The empirical CDF of the rolling coefficient of variation (1-second window) indicates that median load varies by 4.7–5.2 standard deviations from the mean, demonstrating highly dynamic traffic conditions.

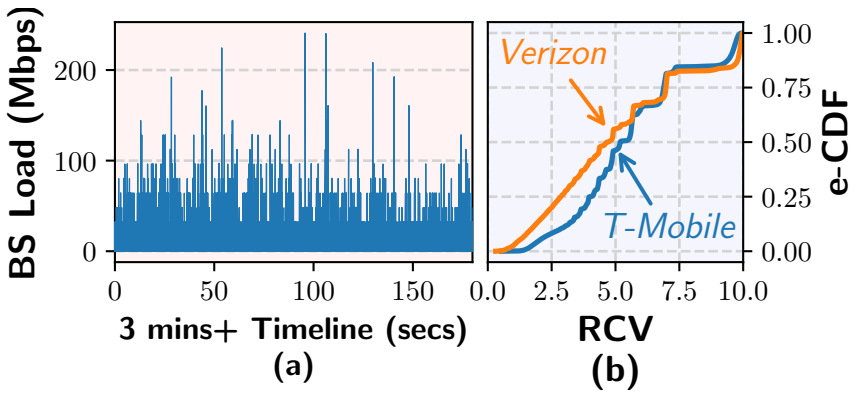


Figure 4.1: The high variability in traffic load observed from two public 5G networks.

(ii) Static operator configurations. Despite such variability, current networks employ static TDD policies. Table 4.2 shows that major operators use fixed slot configurations independent of load conditions, even though 3GPP allows multiple configurable TDD patterns.

Table 4.2: TDD policies used by major public 5G operators in the U.S.

Operator	Band	Pattern 1 (UL/DL slots)	Pattern 1 (UL/DL symbols)	Pattern 2 (UL/DL slots)	Pattern 2 (UL/DL symbols)
T-Mobile	n41	2/3	4/4	0/4	0/0
Verizon	n77	2/3	4/6	0/4	0/0
AT&T	n77	2/3	4/6	0/4	0/0

(iii) DL-biased allocations. Traffic is predominantly downlink-heavy (median DL load 12.9 times UL). Consequently, operators allocate most symbols to DL (e.g., 72.8% DL allocation in T-Mobile). However, emerging applications increasingly require UL bandwidth and low latency, making DL-biased configurations suboptimal.

(iv) Non-negligible UL-heavy periods. Although DL dominates overall, 4.1% of observed intervals exhibit higher UL load than DL. These periods highlight the need for dynamic TDD adjustment rather than static allocation.

Case study. We evaluate a live video ingest application on the over-the-air testbed using five TDD configurations (S1–S5). Static UL-heavy allocations improve QoE initially, but excessive UL

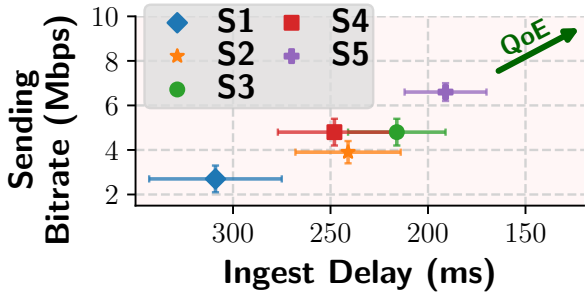


Figure 4.2: Impact of static TDD policies on the live video ingest application's QoE (sending bitrate and ingest delay).

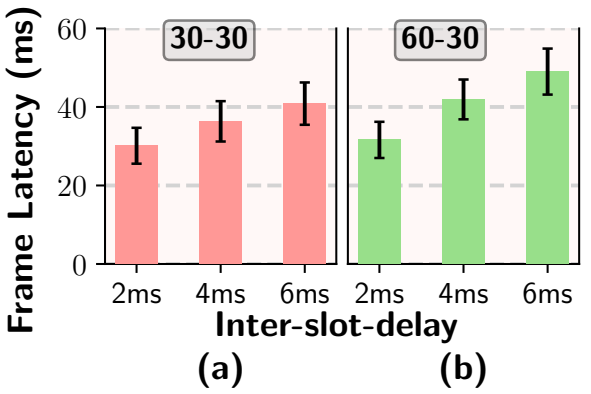


Figure 4.3: Effect of *inter-slot delay* on Edge Video Analytics (EVA) frame response latency under different background traffic settings.

allocation degrades DL performance due to limited resources. In contrast, our dynamic approach adapts to load changes, achieving 37.5% higher bitrate and 11.6% lower ingest delay than the best static configuration (Figure 4.2).

Reactive policies are insufficient. A naive dynamic policy that adjusts slot ratios based on past traffic load performs poorly because traffic conditions change rapidly. Reactive approaches therefore lag behind optimal configurations and result in measurable QoE loss.

Slot arrangement also matters. Beyond UL/DL ratios, the ordering of slots within a TDD pattern affects latency-sensitive workloads. Different arrangements lead to different inter-slot delays. For edge video analytics workloads, increasing inter-slot delay from 2 ms to 6 ms increases frame latency by 35.5% (Figure 4.3). This demonstrates that both slot distribution and arrangement must be jointly optimized.

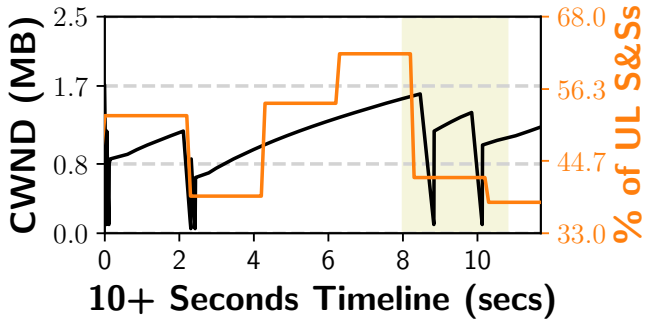


Figure 4.4: Impact of frequent TDD policy updates on TCP congestion control behavior.

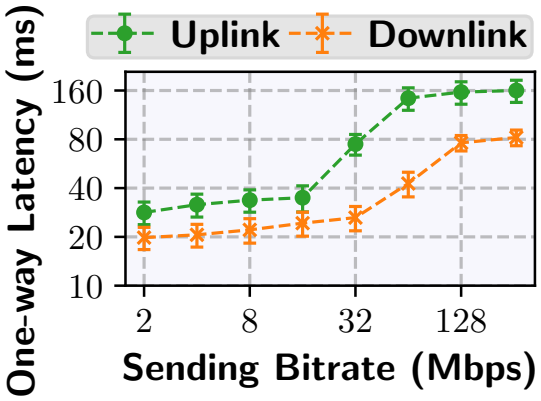


Figure 4.5: Comparison of one-way UL and DL latencies in the testbed (log-scale).

4.2.3 Challenges

Scalability. The flexibility of NR allows thousands of possible UL/DL slot arrangements. For numerology $\mu = 1$, there are more than 1450 possible configurations. Exhaustively searching this space in real time is computationally prohibitive.

Highly dynamic environment. Traffic load, application demands, and channel conditions change rapidly. Static or purely reactive solutions fail to track these dynamics. Furthermore, the BS lacks explicit QoE feedback from applications, requiring indirect optimization via radio-layer metrics.

Interference with higher-layer adaptation. Frequent TDD policy changes can destabilize transport-layer congestion control and application-level rate adaptation. For example, abrupt UL slot reduction triggers TCP timeouts and congestion window collapse (Figure 4.4), leading to sustained throughput loss.

Asymmetric UL and DL transmission. Our measurements reveal a clear asymmetry between UL and DL performance, not only in throughput but also in one-way latency. Figure 4.5 shows results from our over-the-air testbed: even with equal UL/DL slot allocation, UL one-way

latency is roughly 40% higher than DL at low sending rates, and UL latency increases sharply (e.g., due to bufferbloat or constrained UL radio resources) once the offered load exceeds link capacity. Public-network traces in D3 exhibit the same trend. Contributing factors include limited UE transmit power [56], delays from UL scheduling grants [106], reduced carrier aggregation on UL [124], and the use of SC-FDMA for power efficiency [25].

4.3 Design Overview

To address the challenges outlined above (§4.2.3), Wixor employs a *two-stage* approach to TDD policy adjustment. First, it predicts the UL and DL S&S distribution (percentages) based on the BS context such as traffic load and channel quality. Once the distribution is determined, Wixor finds the *best* S&S arrangement. This decomposition significantly reduces the search space compared to an exhaustive search method, evaluating only 5–25 arrangements for $\mu = 1$ (a 58–290$\times$ reduction). While this two-stage approach may incur slight performance losses if the initial prediction is inaccurate—especially when relying on fixed models that do not generalize well to complex RAN environments—Wixor mitigates this risk by employing a *learning-based* approach in combination with BS-level features. This approach effectively manages the complexity of the environment and the asymmetry between UL and DL transmission. Additionally, the system utilizes a conservative policy smoothing technique to prevent abrupt policy changes, thereby minimizing interference with transport-layer congestion control and application-layer rate adaptation logic.

The basic operation of Wixor is illustrated in Figure 4.6. In the UL direction, UEs request radio resources from the BS, obtain the allocated UL resources, and transmit data that is forwarded to the Internet. Conversely, in the DL direction, incoming data arrives in per-UE queues, and the

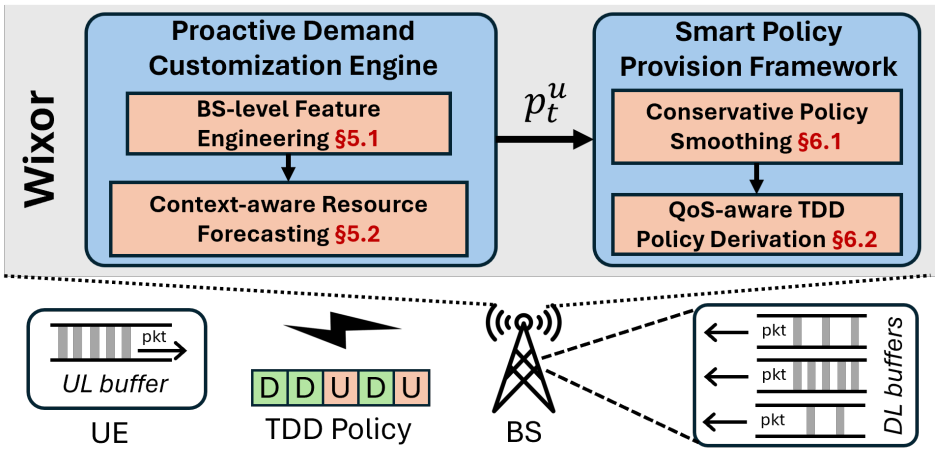


Figure 4.6: A high-level overview of Wixor.

BS schedules transmissions to UEs. In this process, ensuring that the BS effectively *balances* available TDD S&Ss between UL and DL such that UEs receive *sufficient* resources *promptly* is key to improving application QoE. Wixor operates as a lightweight service at the BS to enable timely TDD policy adjustment.

To this end, Wixor leverages traffic demand, BS load, channel quality, and QoS features—readily available at the BS—as system inputs, and outputs a TDD pattern that guides BS TDD policy adjustment through two major modules:

(i) A **proactive demand customization engine** (§4.4) that predicts future UL and DL resource demands. It utilizes cross-layer BS-level features to capture the RAN context (§4.4.1). These features are fed into a context-aware resource forecasting module (§4.4.2), which outputs S&S percentage allocations for UL and DL.

(ii) A **smart policy provision framework** (§4.5) that ensures reliable TDD policy configuration for QoE improvement. It first applies conservative policy smoothing to reduce the impact of abrupt TDD changes on application QoE (§4.5.1). Then, the QoS-aware TDD policy derivation module (§4.5.2) computes the final arrangement of UL and DL S&Ss within the TDD pattern. In

doing so, it balances the trade-off between inter-slot delay (which impacts latency) and guard period overhead (which impacts throughput).

4.4 Proactive Demand Customization

Wixor first constructs BS-level features from raw BS logs (§4.4.1). These features are then passed to a reinforcement learning (RL) agent to forecast future traffic demand (§4.4.2). The RL agent employs a neural network (NN) to interpret the RAN context represented by the BS-level features. Training an RL agent directly in a live 5G environment is impractical due to the exploration required by RL, which would negatively impact application QoE. Therefore, we train Wixor's RL agent using a faithful simulator with real-world traffic and channel traces (§4.2.1). To ensure transferability from simulation to deployment, all features are normalized prior to model training.

4.4.1 Cross-layer BS-level Feature Engineering

We considered two feature representations: per-UE features and aggregated BS-level features. Per-UE features capture fine-grained user behavior but present practical challenges. First, the number of active users n_t varies over time, requiring variable-sized NN inputs and reducing scalability. Second, input dimensionality grows with n_t, leading to the curse of dimensionality and reduced learning effectiveness.

Instead, Wixor aggregates per-UE measurements into fixed-size BS-level feature vectors using statistical summaries that retain essential information while remaining scalable.

(i) Traffic demand features. Traffic demand features capture the instantaneous UL and DL demand of all active users. We construct the feature vector

$$\mathcal{D}_t = \{B_t^u, B_t^d, M_t^u, M_t^d, A_t^u, A_t^d, H_t^u, H_t^d\}$$

where B denotes average buffer occupancy, M the maximum buffer level, A the data arrival rate, and H the head-of-line delay. These metrics are computed by aggregating per-UE RLC queue measurements and normalized by buffer capacity or BS throughput limits to ensure consistency across deployments.

(ii) BS load features. We incorporate load features

$$\mathcal{L}_t = \{T_t^u, T_t^d, R_t^u, R_t^d\}$$

where T represents UL/DL throughput normalized by maximum BS throughput, and R represents UL/DL resource utilization based on assigned resource blocks.

(iii) Channel quality features. Channel conditions strongly affect achievable throughput and optimal TDD allocation. Rather than averaging CQI values, which masks heterogeneity across UEs, we use percentile statistics:

$$\mathcal{C}_t = \{\text{P25}(c^u), \text{P25}(c^d), \text{P50}(c^u), \text{P50}(c^d), \text{P75}(c^u), \text{P75}(c^d)\}$$

CQI values are normalized by the maximum CQI value (31) to enable model transferability.

(iv) QoS tolerance feature. Finally, a buffer tolerance factor $\rho_t \in [0, 1]$ represents the BS's tolerance to buffering delay. Lower ρ_t corresponds to latency-sensitive workloads.

4.4.2 Context-aware Resource Forecasting

We consider two possible optimization objectives: directly optimizing application QoE, or optimizing radio-layer QoS metrics that correlate with QoE. Because explicit QoE feedback is not available at the BS, Wixor optimizes QoS objectives:

- **O1:** maximize UL and DL throughput, $\max T_t^u, \max T_t^d$
- **O2:** minimize latency proxy via maximum buffer occupancy, $\min M_t^u, \min M_t^d$
- **O3:** minimize buffer overflow risk, $\min(1 - M_t^u), \min(1 - M_t^d)$

Reward. The RL reward combines these objectives:

$$r_t = \eta(T_t^u + \rho_t - M_t^u) + (1 - \eta)(T_t^d + \rho_t - M_t^d) \tag{4.1}$$

where η controls UL priority and ρ_t balances throughput vs latency sensitivity.

State. At time t, the RL state is

$$s_t = \{\mathcal{D}_{t-k:t}, \mathcal{L}_{t-k:t}, \mathcal{C}_{t-k:t}, \rho_t\}$$

capturing recent traffic demand, load, channel quality, and QoS tolerance.

Action. The RL agent outputs UL slot percentage $p_t^u \in [0, 1]$, with DL and guard allocations derived later.

RL training. We train the agent using Soft Actor-Critic (SAC) due to its stability and sample efficiency in networked systems. Multiple BS simulators run in parallel and asynchronously update

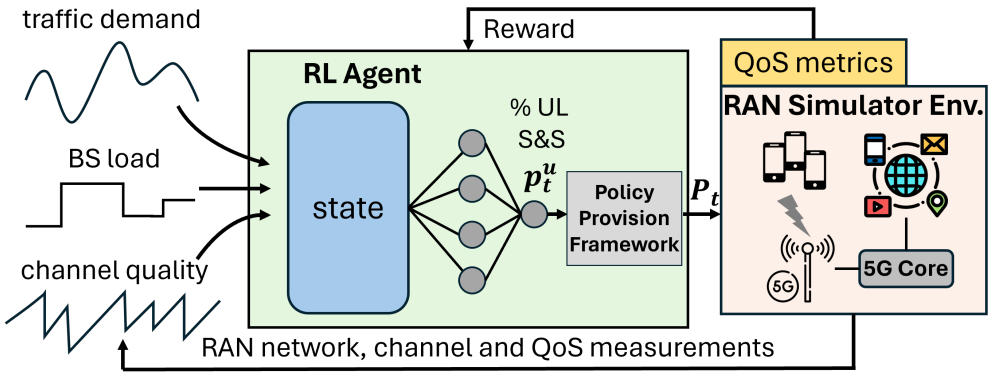


Figure 4.7: Context-aware resource forecasting using reinforcement learning.

a central actor network. At runtime, exploration is disabled and the trained policy is updated periodically using recent BS logs with a small learning rate to adapt to evolving traffic patterns.

4.4.3 RL Agent Architecture and Training

Figure 4.7 illustrates the RL interaction loop between the simulated environment and the agent. The actor network represents the policy $\pi_\theta(a_t|s_t)$ using a neural network. After applying each action, the simulated environment provides the learning agent with a reward r_t. The RL agent continually performs gradient descent to improve the RL policy.

To further improve and accelerate training, Wixor launches multiple RL agents to operate concurrently. By default, we employ 8 parallel agents. Each agent is set up with different input parameters (*e.g.,* channel traces and traffic workloads). These agents continuously transmit their {state, action, reward} tuples to a central agent, which aggregates the data to create a unified model. For each received sequence of tuples, the central agent employs the SAC algorithm to compute policy gradients and perform gradient descent. Subsequently, the central agent updates the actor network and distributes the updated model to the corresponding agent that sent the

tuple. This process occurs asynchronously among all agents, eliminating the need for a locking mechanism between them.

Neural network design. Each feature group is first embedded using a 1D CNN layer (kernel=4, channels=64). QoS features use a fully connected layer. Extracted features are concatenated and passed through additional fully connected layers, producing a 64-dimensional representation. The final output layer (sigmoid activation) produces UL percentage p_t^u. We implement the model using TensorFlow with Adam optimizer, learning rate 10^{-3}, batch size 64, and history length $k = 8$. TensorFlow Serving is used for deployment at the BS.

4.5 Context-aware Policy Provision

After receiving the predicted UL S&S percentage from the proactive demand customization stage, Wixor applies a conservative policy smoothing technique to reduce abrupt policy changes (§4.5.1). It then balances the tradeoff between inter-slot delay and guard period overhead to determine the best TDD pattern (§4.5.2).

4.5.1 Conservative Policy Smoothing

As noted in §4.2.3, abrupt TDD policy changes caused by fluctuating load can mislead transport-layer congestion control and application-layer rate adaptation. Wixor therefore needs to tolerate short-term traffic noise while still reacting promptly to long-term shifts in demand (cf. Figure 4.1a). To achieve this, Wixor applies a conservative smoothing filter to the action $a_t = p_t^u$ produced by the resource-forecasting module:

$$\gamma_t = \beta\,\gamma_{t-1} + (1-\beta)\,|p^u_t - p^u_{t-1}| \tag{4.2a}$$

$$\alpha_t = \gamma^t_b \;/\; \max(\gamma_{t-t_s:t}) \tag{4.2b}$$

$$\hat{p}^u_t = \alpha_t p^u_t + (1-\alpha_t)\hat{p}^u_{t-1} \tag{4.2c}$$

Eqn. 4.2c is an Exponentially Weighted Moving Average (EWMA) that produces the smoothed UL-percentage $\hat{p}^u_t$. We make the EWMA weight time-varying: Eqn. 4.2a computes a smoothed measure γ_t of recent policy variation (we use $\beta = 0.5$ empirically), and Eqn. 4.2b normalizes the recent variation across a larger window $[t - t_s, t]$ (we use $t_s = 30\Delta t$). Intuitively, a sudden large change in the predicted UL percentage yields a large α_t, making $\hat{p}^u_t$ rely more on the fresh prediction p^u_t and less on the previous smoothed value. The result is a scheme that damps short erratic predictions while allowing quick response to sustained shifts.

4.5.2 QoS-aware TDD Policy Derivation

Given the smoothed UL percentage $\hat{p}^u_t$, Wixor must choose an actual TDD policy $\mathcal{P}_t$ (i.e., a specific arrangement of UL, DL, and guard symbols) that respects guard-period constraints and balances latency versus throughput. Deriving $\mathcal{P}_t$ is non-trivial because the arrangement determines the *inter-slot delay* (time between consecutive UL or DL opportunities) which strongly affects latency-sensitive workloads, while every direction switch incurs guard symbols that reduce effective throughput.

Guard period calculation. Guard periods are inserted to avoid UL/DL interference and to give transceivers time to switch between Tx/Rx modes. The required guard symbols depend mainly on

two factors: BS coverage radius (which sets propagation delay) and transceiver switching delays (BS and UE hardware). Wixor computes the number of guard symbols for DL→UL transitions ($g^{d,u}$) and UL→DL transitions ($g^{u,d}$) using the per-symbol duration Δs determined by numerology μ and cyclic prefix Δcp:

$$\Delta s = \frac{1}{2^{\mu} \times 15\text{kHz}} + \Delta cp \tag{4.3a}$$

$$g^{d,u} = \left\lceil \frac{2 \times \mathcal{R}/c + \Delta_{ue}^{Rx,Tx}}{\Delta s} \right\rceil \tag{4.3b}$$

$$g^{u,d} = \left\lceil \frac{\Delta_{bs}^{Rx,Tx}}{\Delta s} \right\rceil = \left\lceil \frac{N_{TA,Offset} \cdot T_c}{\Delta s} \right\rceil \tag{4.3c}$$

Here $\mathcal{R}$ is the maximum BS coverage radius, c is the speed of light, $\Delta_{ue}^{Rx,Tx}$ is the UE Rx→Tx switching delay, and $\Delta_{bs}^{Rx,Tx}$ (expressed via $N_{TA,Offset} \cdot T_c$) captures BS switching/timing offset. As an example, for $\mu = 1$, $\mathcal{R} = 100\,\text{m}$ and $\Delta cp = 2.34\mu$s, the formulae suggest $g^{d,u} = 2$ and $g^{u,d} = 1$ guard symbols in typical settings.

Computing valid TDD policy set. Given $\hat{p}_t^u$, $g^{d,u}$ and $g^{u,d}$, Wixor enumerates all legal S&S arrangements that (i) respect 3GPP TDD constraints and (ii) satisfy the target UL/DL/guard percentages (within rounding). For a practical range of numerology and periodicity values, $\mathcal{S}_t$ (the candidate set) typically contains 5–25 valid patterns. For each candidate pattern $s \in \mathcal{S}_t$ we compute:

- $p^{g,s}$ — the guard-period overhead (fraction of symbols used for guard intervals), and
- d^s — the aggregate inter-slot delay (sum or appropriate metric of DL→UL and UL→DL inter-slot delays).

Guard overhead in common patterns ranges from roughly 0.2% to 3.1% of available symbols depending on switching and propagation conditions.

Selecting the best TDD policy. Wixor balances the tradeoff between latency (inter-slot delay) and throughput loss (guard overhead) via the buffer-tolerance factor ρ_t introduced earlier. We compute a normalized per-pattern weight

$$w^s = \rho_t \frac{d^s}{\sum_{s \in \mathcal{S}_t} d^s} + (1 - \rho_t) \frac{p^{g,s}}{\sum_{s \in \mathcal{S}_t} p^{g,s}} \tag{4.4}$$

and choose the pattern with minimal w^s:

$$\mathcal{P}_t = \arg\min_{s \in \mathcal{S}_t} w^s.$$

Lower ρ_t biases toward throughput (smaller guard overhead), while higher ρ_t prioritizes lower inter-slot delay (better latency).

Design notes. This policy derivation is heuristic rather than provably optimal, primarily because explicit per-application QoE feedback is not available at the BS. Nonetheless: (i) the approach works well in realistic traces and over-the-air tests (§4.7.5), (ii) it gives operators a clear knob ρ_t to trade latency versus throughput, and (iii) the ρ_t computation can be replaced with any other mapping (e.g., learned mapping from historical QoE estimates). In our implementation we provide two ρ_t variants: a fixed default and a dynamic default that derives ρ_t from the count of latency-sensitive flows present at the BS (implementation detail in §4.6).

4.6 Wixor Implementation

Wixor prototype. Wixor is built on top of srsRAN [158, 129], an open-source 5G software-defined radio suite. We modified the user-plane protocol stack (5G Layer 2) in srsRAN to implement Wixor in over 2.3K lines of C/C++ code. First, we added support for dynamic TDD, enabling runtime TDD policy adaptation. We then implemented logging functionality for the PDCP, RLC, and MAC layers to support feature engineering.

To support dynamic policy adjustment, we developed a modular TDD policy adaptation engine on top of the TDD MAC scheduler. This engine supports arbitrary TDD policies and receives BS logs at configurable periodic intervals. It exposes a callback interface that allows runtime updates to the TDD pattern. Wixor is implemented as a derived class of this modular engine. It processes BS logs to construct BS-level features (§4.4.1), which are then passed to Wixor's RL agent (§4.4.2). Deployed using TensorFlow Serving [148], the RL agent outputs the UL S&S percentage, which is post-processed using the conservative policy smoothing technique (§4.5.1). Wixor then derives the best TDD pattern using the UL S&S percentage and guard period information (§4.5.2).

If the newly computed TDD pattern differs from the current configuration, Wixor waits until the next transmission period before applying the update through the callback interface. Once triggered, the modular adaptation engine updates the BS's internal data structures that maintain TDD configuration. We believe this deployment is practical, given the increasing programmability of modern cellular base stations [35].

Faithful simulator. We developed a faithful 5G network simulator based on the ns-3 5G-Lena [16] codebase. The simulator mirrors the over-the-air prototype, including support for dynamic TDD and the modular policy adaptation engine. We integrated trace-driven channel

simulations and implemented application traffic workload generators as described in §4.2.1. To train the RL models, we used the ns3-gym toolkit [63] together with TensorFlow [21]. Overall, we added or modified more than 4.2K lines of C/C++ and Python code.

Data collection for feature engineering. Recall from §4.4.1 that Wixor relies on traffic demand, BS load, channel quality, and QoS features to characterize the RAN context. Specifically, DL buffer occupancy $b_t^{d,i}$ for UE i is obtained from the RLC per-UE queues, while UL buffer occupancy $b_t^{u,i}$ is inferred from quantized Buffer Status Reports (BSRs) extracted from MAC control elements. Inter-arrival times $\lambda_t^{u,i}$ and $\lambda_t^{d,i}$ are computed from the arrival of RLC service data units (SDUs) in UL and DL queues, respectively, and average SDU sizes $\hat{s}_t^{u,i}$ and $\hat{s}_t^{d,i}$ are estimated from these queues.

The DL head-of-line delay $h_t^{d,i}$ corresponds to the waiting time of the first SDU packet in the DL RLC queue. UL head-of-line delay $h_t^{u,i}$ is estimated as $(b_{t-1}^{u,i} - t_t^{u,i} \cdot \Delta t)/\hat{s}_t^{u,i}$, where $t_t^{u,i}$ ($t_t^{d,i}$) represents UL (DL) throughput obtained from the PDCP layer, and Δt is the system time step. Resource utilization metrics $r_t^{u,i}$ and $r_t^{d,i}$ are obtained from MAC scheduler allocations. DL channel quality $c_t^{d,i}$ is obtained from CQI reports, while UL CQI $c_t^{u,i}$ is measured directly at the BS.

Finally, the buffering tolerance factor ρ_t can be configured either as a fixed value or derived from application QoS requirements using the 5G QoS Identifier (5QI). By default, Wixor computes ρ_t as the fraction of latency-sensitive flows at the BS. Specifically, each active data bearer is associated with a 5QI value, which encodes latency, reliability, and priority requirements defined by 3GPP. Wixor maps these 5QI values to latency sensitivity and computes ρ_t as the ratio of latency-sensitive flows to the total number of active flows. The mapping between applications and their corresponding 5QI values used in our evaluation is summarized in Table 4.3.

Table 4.3: 5G QoS Identifier (5QI) values used for evaluated applications.

Application	5QI	Latency-sensitive	Application	5QI	Latency-sensitive
EVA	7	Yes	LVC	80	Yes
EAVP	84	Yes	HFT	6	No
LVI	71	No	Background Traffic	6	No
VoD	6	No			

The 5G QoS Identifier (5QI) specifies QoS requirements such as packet delay budget, packet error rate, and scheduling priority for each data bearer. In our implementation, each application establishes data bearers with the corresponding 5QI value shown in Table 4.3. These values are used to determine whether a flow is latency-sensitive and therefore influence the buffering tolerance factor ρ_t used in TDD policy derivation.

In simulation experiments, applications explicitly configure data bearers with their assigned 5QI values, enabling Wixor to compute ρ_t dynamically. However, commercial smartphones (e.g., Pixel 7) do not expose APIs to configure bearer-level 5QI. Therefore, in over-the-air experiments we use a fixed value of $\rho_t = 0.5$ to represent a balanced latency-throughput tradeoff.

4.7 Evaluation

4.7.1 Experiment Setup

Methodology. Our experiments use 5G numerology μ=1 (30 KHz subcarrier spacing) and the proportional fair MAC scheduler, unless otherwise mentioned. The BS operates with Band 78 @ 3410 MHz and 20 MHz channel bandwidth. The channel quality indicator (CQI) reporting interval is set to 40 ms. We train Wixor with 60% of the (channel and traffic) traces and use the rest for evaluation. The UL and DL priority for Wixor is equal (*i.e.,* η=0.5), and the buffering tolerance factor (ρ_t) is adjusted according to the 5QI method discussed in §4.6.

Baselines. ***(i)*** *Default:* the default static, DL-heavy TDD policy employed by current 5G networks (*i.e.,* 22.8% UL and 72.8% DL S&Ss); ***(ii)*** *SFair:* a static TDD policy that fairly distributes S&Ss among UL and DL based on the average traffic load of an experiment; ***(iii)*** *Reactive:* a TDD policy that configures S&Ss at time step t according to the previous time step's traffic load; ***(iv)*** *DRP [29]:* a recent RL-based algorithm that derives UL and DL S&S percentage to minimize buffer overflows. To the best our ability, we train *DRP*'s RL agent using BS-level features and parameters described in the paper. Further, we use the same UL and DL priority as Wixor.

Applications and metrics. Apart from the trace-generated background traffic (§??), we use six diverse application workloads to generate traffic. ***(i)*** *Edge Video Analytics (EVA)*: We select a popular EVA task, *i.e.,* Object Detection. The EVA app uses a state-of-the-art video analytics model (*i.e.,* YOLOv7 [175]) deployed on the edge server. Instead of sending camera feeds, a UE streams video frames from the COCO dataset [98] at 30 FPS. ***(ii)*** *Edge-assisted Vehicle Perception (EAVP)*: Autonomous vehicles rely on object tracking to ensure safe and robust driving performance. Using siamFC++ model [185], we set up an EAVP app on the edge server for multiple object tracking. The UE transmits five camera feeds (front and sides) at 30 FPS using the Waymo Open Dataset [162]. ***(iii)*** *Live Video Ingest (LVI)*: We re-purpose Ant-Media's LiveVideoBroadcaster [23] to publish a pre-recorded video stream (1080p @ 30 FPS with 6.5 Mbps average bitrate). The UEs send adaptive RTMP feeds to an Ant Media server deployed on the application server. ***(iv)*** *Video-on-Demand (VoD)*: Our VoD streaming experiments use a dash.js player to stream a 4 min video. We mainly test buffer-based BOLA and rate-based adaptive bitrate (ABR) algorithms. The video is encoded at 6 quality levels with average bitrates ranging from 0.8 Mbps to 6 Mbps. ***(v)*** *Live Video Conferencing (LVC)*: We implement a peer-to-peer LVC app based on WebRTC. Instead of using the camera, the LVC app streams a 1280 $\times$ 780 pre-recorded meeting

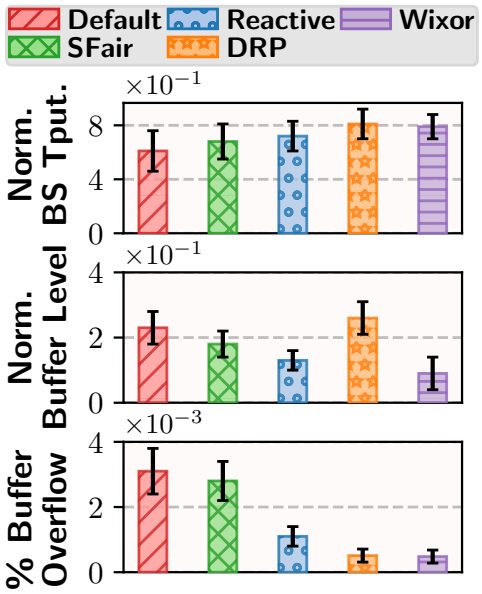


Figure 4.8: BS QoS metrics for the simulation experiments in Table 4.4.

video at 30 FPS. ***(vi)*** *HTTP File Transfer (HFT)*: The UE repeatedly uploads/downloads a 128 MB file to/from the application server. We log the total file download/upload time to show results.

4.7.2 Overall Benefit for the Applications

We conduct extensive simulations to evaluate the performance of Wixor. Our evaluation only utilizes the six applications described earlier (§4.7.1) to generate user traffic; we do not generate any background traffic for this experiment. Each app has up to 10 instances running concurrently, with each instance running for a maximum of 300 secs. To distribute the traffic temporally, we generate app instance start times using a Poisson random process. Specifically, we determine inter-arrival times for app instances with an arrival rate $\lambda = 10/300 = 0.033$ and convert these to start times. Each experiment ran for 30 mins, using random channel traces from our *corpus*. We repeated each experiment 5×, selecting different random traces for each run. The overall traffic load for the experiment ranged between 30% and 90%.

Table 4.4: Overall QoE comparison across six application workloads over 6+ hours of channel traces. Values denote mean ± standard deviation. Arrows indicate improvement direction; shaded cells denote best performance.

Application	Metric	Default	SFair	Reactive	DRP	Wixor
Edge Video Analytics (EVA)	Response Latency (ms) ↓	93.3±10.6	77.2±9.1	44.9±7.5	57.4±7.1	38.1±6.4
	Perceptive Accuracy (%) ↑	34.7±6.2	40.1±6.1	54.8±7.0	47.6±7.9	68.2±8.5
Edge-assisted Autonomous Vehicle Perception (EAVP)	Response Latency (ms) ↓	67.8±6.5	60.3±6.5	51.7±6.1	56.8±6.2	46.3±5.9
	Mean IoU ↑	0.68±0.1	0.71±0.1	0.77±0.2	0.74±0.1	0.78±0.1
Live Video Ingest (LVI)	Ingest Delay (ms) ↓	284.9±34.7	255.3±28.5	246.4±28.3	233.8±24.0	191.5±22.5
	Sending Bitrate (Mbps) ↑	3.8±1.5	5.5±1.2	5.8±1.3	6.1±1.3	6.2±1.2
Video-on-Demand (VoD) Streaming	Normalized Bitrate ↑	0.83±0.1	0.63±0.2	0.77±0.2	0.80±0.2	0.81±0.2
	Stall Percentage (%) ↓	0.63±0.3	0.11±0.1	0.18±0.1	0.25±0.2	0.12±0.2
Live Video Conferencing (LVC)	Video Quality (SSIM dB) ↑	15.3±1.1	13.1±1.4	14.1±1.5	14.6±1.1	15.7±1.4
	Video Delay (ms) ↓	48.7±5.6	65.9±5.8	48.3±5.7	58.4±6.3	43.9±5.2
HTTP File Transfer (HFT)	Upload Time (s) ↓	562.3±75.7	422.9±64.3	418.4±74.7	397.3±63.9	405.7±61.4
	Download Time (s) ↓	352.8±56.9	379.8±54.2	376.3±58.4	381.6±62.7	385.2±60.3

Overall QoS improvement. Wixor considers three BS QoS metrics in its overall objective for all applications, *i.e.,* BS throughput, buffer level, and buffer overflows (§4.4.2). Figure 4.8 compares these metrics across all baselines to Wixor. There are two main takeaways: *(i)* Wixor outperforms static TDD policies (*Default* and *SFair*) across all metrics. For instance, it achieves an average 16.1%-29.5% higher BS throughput compared to static schemes. While *Default* provides high DL throughput, its UL performance degrades due to the DL-heavy S&S allocation; and *(ii)* *DRP* performs similarly to Wixor in terms of BS throughput and buffer overflows. However, *DRP* maintains 1.9× higher average buffer level than Wixor. Since *DRP*'s reward function prioritizes a high buffer level while avoiding overflows, latency-sensitive applications will experience significant performance degradation.

QoE benefits. Table 4.4 showcases the overall QoE gains Wixor brings for various applications. Our results highlight three main findings: *(i)* Wixor achieves significantly higher performance than the baselines for all latency-sensitive applications (EVA, EAVP, and LVC). For EVA, it achieves an average 24.4%-96.5% higher perceptive accuracy compared to other schemes,

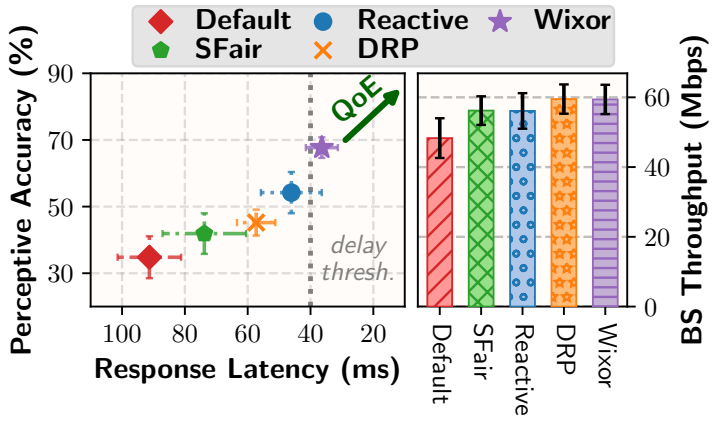


Figure 4.9: Comparison of Edge Video Analytics QoE across baselines.

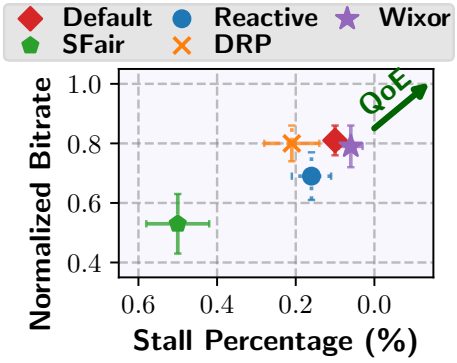


Figure 4.10: Comparison of Video-on-Demand streaming QoE across baselines.

while reducing the response latency by 15.1%-59.2%. The reduction in latency can be mainly attributed to Wixor's latency-aware optimization objective (§4.4.2) and policy derivation mechanism (§4.5.2). *(ii) Default*, having DL-heavy S&S allocation, performs slightly better than Wixor for the DL bandwidth-intensive applications (VoD, HFT download). As an example, *Default* has an average 2.5% and 3.7% higher VoD bitrate than Wixor and *DRP*, respectively. *(iii)* Wixor and *DRP* offer similar performance for the UL bandwidth-intensive applications (HFT upload, LVI). *To summarize,* Wixor primarily balances UL and DL S&S allocation to promptly provision resources for all application types - it outperforms or is within 91.6% of baselines for all metrics. Note that here we used Wixor's default parameter values (*e.g.,* η=0.5). We later (§4.7.5) show that Wixor can be easily tuned to prefer certain use cases.

4.7.3 Over-the-air Evaluation of Wixor

We evaluate Wixor prototype with a combination of application and trace-generated background traffic. We conduct 5 hrs+ of experiments using the over-the-air setup described in §4.2.1. The background traffic is generated with 15 random UE traces from D3.

Table 4.5: System overhead at 65% traffic load.

Metric	Default	Wixor	Δ (abs)
CPU utilization (%)	52.4 ± 6.5	60.1 ± 6.8	7.7
Memory utilization (%)	23.9 ± 2.1	26.2 ± 2.0	2.3

Table 4.6: NN inference time (mean ± std).

Method	Inference time (ms)
CPU	13.1 ± 1.9
GPU	6.8 ± 1.3

Dissecting Wixor's performance gains. We first test the EVA app that runs on one PX7, with background traffic running on the other. These experiments use a fixed ρ_t (*i.e.,* 0.5). Figure 4.9 (left) shows the frame response latency and perceptive accuracy. In our setup, the latency-sensitive EVA requires a frame response latency of less than 40 ms, as indicated by the dotted gray line. To demonstrate that Wixor maintains fairness for other users, we also plot the overall BS throughput (right). There are four key takeaways. *(i)* Wixor significantly outperforms the baselines across both metrics. For example, it offers an average 24.9%-94.5% higher perceptive accuracy and 21%-60.1% lower response latency than baselines. *(ii)* Unlike *DRP*, Wixor's incorporation of latency objectives in its RL reward (Eqn. 4.4.2) minimizes buffering delays for latency-sensitive applications. On average, *DRP* incurs 36.5% higher response latency than our approach, largely due to *DRP*'s higher buffer levels and queuing delays. Although not shown here, *DRP*'s median buffer level is 1.4× larger than Wixor's during the experiment. *(iii)* Static schemes like *Default* and *SFair* cannot adapt to changing traffic loads, leading to the lowest performance in our tests. *(iv)* Interestingly, *Reactive*, by following the UL and DL traffic patterns to reduce buffer levels, outperforms *DRP* in terms of QoE, delivering 19.6% and 19.9% better average response latency and perceptive accuracy, respectively. However, *Reactive* still falls short of Wixor performance due to its reactive nature (§4.2.2).

Next, we evaluate Wixor with the most ubiquitous form of Internet traffic, *i.e.,* VoD streaming. Our results in Fig. 4.10 indicate that Wixor, *Default* and *DRP* offer similar throughput performance

for DL bandwidth-intensive apps. For instance, Wixor offers 2.5% lower average bitrate than *Default* while reducing average stall up to 40%. *Default*'s high DL performance comes at the cost of lower UL performance as seen earlier in Table 4.4. *DRP*, on the other hand, performs well for bandwidth-intensive applications, but incurs QoE loss for latency-sensitive applications, as seen above.

Wixor's overhead. We record the CPU overhead and memory consumption of Wixor in Table 4.5. Compared to *Default*, Wixor increases the absolute CPU and memory utilization by 7.7% and 2.3%, respectively. The overhead primarily comes from Wixor's RL agent's resource forecasting (§**??**), which may slightly rise with the number of users. Although not shown here, the CPU and memory utilization only increases by 3.2% and 1.1%, respectively, when the traffic load increases from 65% to 90%. We also compute RL agent's inference time in Table 4.6. By default, Wixor performs inference on a CPU which takes only 13 ms on average. The inference time can be further reduced with a GPU, *e.g.,* NVIDIA GeForce RTX 3060 Ti GPU cuts the average inference time to 7 ms (48.1% reduction).

4.7.4 Wixor under Diverse Settings

We use *ns-3* simulations, with trace-generated background traffic (from 30 random UE traces unless otherwise mentioned), to evaluate how Wixor performs under different settings.

Traffic load. We pick three traffic traces with varying fluctuation levels (standard deviation) from D3: ***T1***, ***T2***, and ***T3*** with approximately 60%±10%, 60%±20%, and 60%±30% average traffic load, respectively. Each trace is 10 mins long. Our results in Table 4.7 highlight that Wixor's TDD policy adapts well to the changing load, while other baselines cannot. For example, the absolute

Table 4.7: Average BS throughput gap (%) between Wixor and baselines.

Trace	Δ SFair	Δ Reactive	Δ DRP
T1	9.1	7.3	3.8
T2	12.0	10.2	5.2
T3	15.2	14.9	6.4

Table 4.8: Wixor performance across common 5G numerologies (μ).

μ	Reactive		Wixor	
	BS Throughput	Per-packet Latency	BS Throughput	Per-packet Latency
0	52.2±6.3	48.1±7.0	57.5±6.8	40.1±4.9
1	49.8±6.3	47.4±5.9	55.1±6.7	36.7±5.3
2	48.3±6.4	47.6±6.2	53.0±6.7	33.2±5.2

performance gap between Wixor and *Reactive* increases from 3.5% to 8.5% as load variation goes up.

Radio channel quality. Our channel trace *corpus* consists of various mobility scenarios (*e.g.,* walking and driving). The driving scenario sees more channel fluctuations than the walking case – driving has an average SINR of 14±5 dB while walking has 17±3 dB SINR. Although not shown, Wixor offers higher performance improvement when the radio channel fluctuates frequently due to its use of channel quality features (§4.4.1). For example, Wixor has 1.4% higher BS throughput and 32.1% lower per-packet latency than *DRP* on average for the walking case. For driving, the average throughput and latency improvement is 4.6% (an increase of 3.2%) and 47.4% (an increase of 15.2%), respectively.

Advanced BS configurations. Our evaluations have used a 5G numerology μ=1 (30 KHz subcarrier spacing) so far, which is the μ used by 5G mid-band operators these days [61]. We test Wixor with other numerologies and summarize the results in Table 4.8. In general, higher μ values lead to more slots per subframe which ultimately increases the number of possible S&S arrangements. This results in lower network latency but slightly reduced BS throughput. Wixor therefore reduces network latency when μ increases (or # of S&S arrangements increases).

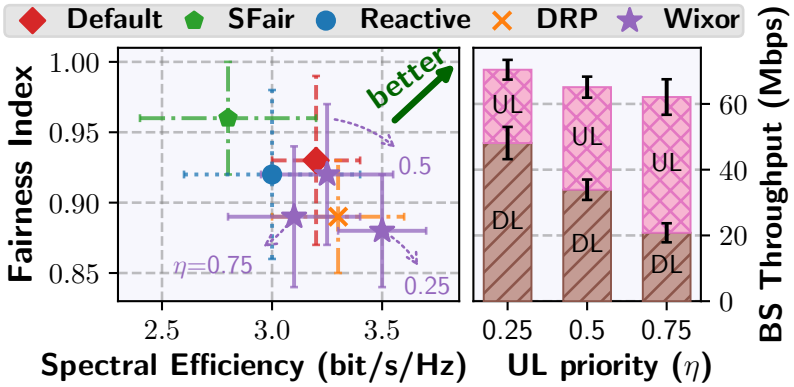


Figure 4.11: Wixor's impact on RAN metrics.

4.7.5 Wixor Deep Dive

RAN objectives. Here, our setup utilizes *ns-3* simulations with trace-generated background traffic only (from 30 random UE traces). Figure 4.11 (left) plots the user fairness and spectral efficiency for different UL priority (η) values. To evaluate user fairness, we calculate the Jain's fairness index of the long-term average throughput among users. Spectral efficiency (bit/s/Hz) indicates the amount of information sent through a network using the available bandwidth. The right plot in Figure 4.11 shows the distribution of UL and DL BS throughput. Our results provide two key insights: *(i)* Wixor's fairness for the default UL priority (*i.e.*, η=0.5) is within 98.9% of the *Default*. In addition, Wixor improves average spectral efficiency by 1.6% compared to *Default*. *(ii)* The η parameter can be adjusted to fine-tune UL performance. A higher η (0.75) enhances UL performance, but the overall BS throughput and spectral efficiency decrease, as UL typically requires more S&Ss to achieve the same performance as DL (§4.2.3).

Scalability and application-level fairness. We evaluate Wixor with a large number of users simultaneously performing HTTP File Transfer (HFT). Each user simultaneously downloads and uploads a 128 MB file. Figure 4.12 plots the average UL and DL file transfer time as the total number of users grow. When users increase, the file transfer time gradually increases due to

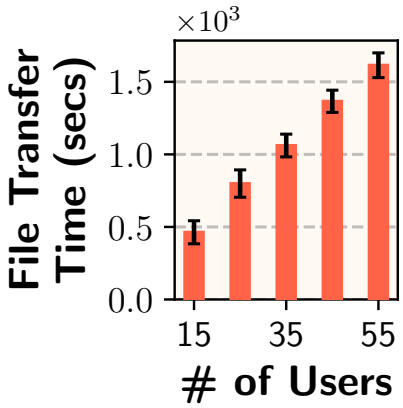


Figure 4.12: System scalability under multiple users.

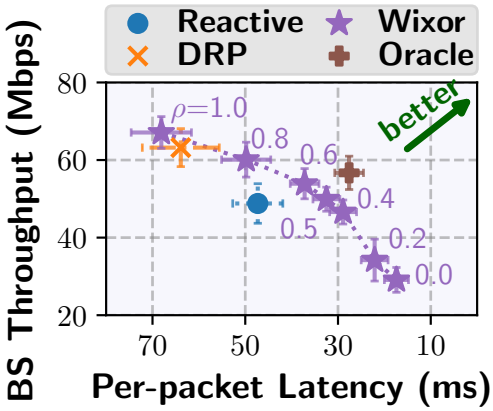


Figure 4.13: Performance gap between Wixor and *Oracle*.

limited bandwidth of our BS. However, the increase is almost linear, and the standard deviations are small, suggesting that Wixor also offers application-level fairness in the presence of multi-user competition.

Optimality. Next, we analyze the performance gap between Wixor and an offline optimal solution (*Oracle*). *Oracle* employs dynamic programming to compute the optimal TDD policy. Specifically, we exhaustively search all TDD patterns to find the one that offers the best performance based on Eqns. 4.4.2 & 4.4 (§4.3). Figure 4.13 illustrates Wixor's BS throughput and per-packet latency for different buffering tolerance factors (ρ) and compares it with *Oracle*. Our results highlight two main findings: *(i)* On average, Wixor is within 82.2% and 88.0% of the *Oracle* for per-packet latency and BS throughput, respectively. The performance gap stems from two factors: prediction errors in the forecasting module (§4.4.2) and performance loss from breaking TDD policy adaptation into sequential steps instead of joint optimization (§4.3). *(ii)* The configurable buffering tolerance factor effectively tradeoffs latency for higher throughput and vice versa.

Prediction accuracy of demand customization engine. We utilize *ns-3* simulations (HFT with 30 users, ρ=0.9) to analyze how well Wixor predicts the UL S&S (p_t^u). Note that we use a high ρ

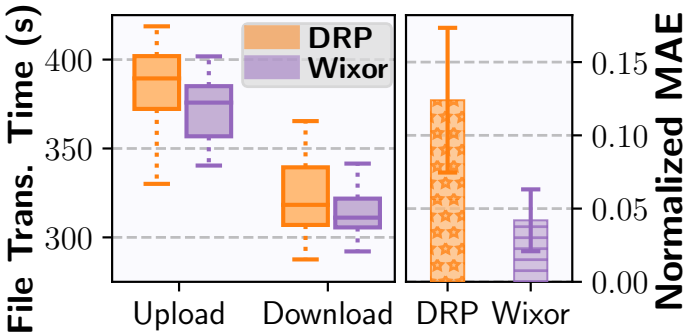


Figure 4.14: Prediction accuracy of the demand customization engine compared with *DRP*.

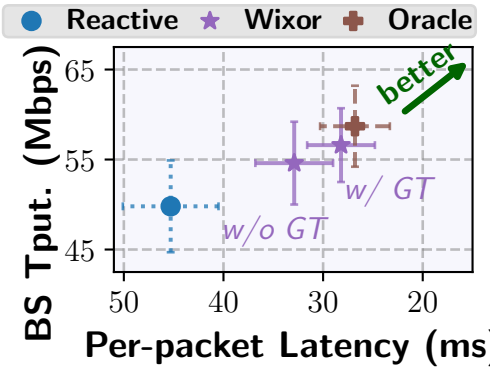


Figure 4.15: Evaluation of the slot derivation module.

value here to tradeoff latency for higher throughput. The left plot in Figure 4.14 quantifies file download and upload times while the right plot shows the Mean Absolute Error (MAE) between p_t^u and the ground truth UL traffic load. Compared to *DRP*, Wixor reduces the median file transfer time by 2.3%-3.5%. Wixor's higher performance can be attributed to its use of cross-layer BS-level features (§4.4.1) and careful learning agent design (§4.5). Overall, Wixor leads to 66.1% lower average MAE than *DRP*.

Contribution of policy derivation module. We investigate if Wixor's policy derivation module (§4.5.2) effectively finds the *best* TDD policy. Again, we use *ns-3* simulations with trace-generated background traffic from 30 random UE traces in D3. We leverage the ground truth (GT) UL traffic load instead of the predicted UL S&S percentage p_t^u for a fair comparison (*w/ GT*). Our results in Figure 4.15 depict that Wixor operates close (3.6%-7.6% gap depending on the metric) to the *Oracle* when it uses ground truth (*w/ GT*) UL S&S percentage. The gap between Wixor and *Oracle* increases slightly (7.0%-25.6%) for the *w/o GT* case due to the p_t^u prediction error.

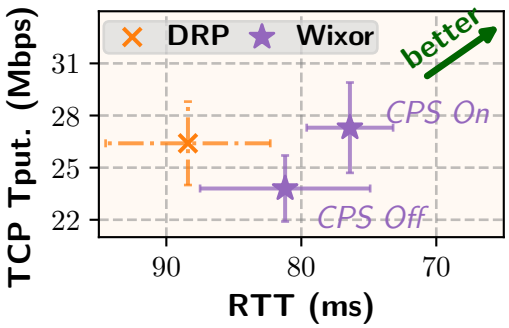


Figure 4.16: Evaluating conservative policy smoothing.

RL Scheme	Average test reward r_t
Wixor's SAC	0.93±0.22
DQN	0.67±0.28
PPO	0.90±0.21

Table 4.9: Comparing different RL schemes.

4.7.6 Micro-benchmarking

Benefit of conservative policy smoothing. Recall from §4.2.2 that abrupt TDD policy changes can mislead rate adaptation modules and result in lost performance. We repeat the same TCP experiment (Figure 4.4) to see how well the conservative policy smoothing (CPS) module addresses the issue. Figure 4.16 compares the TCP throughput and round-trip-time (RTT) for two cases: CPS enabled and CPS disabled. Compared to the case when CPS is disabled, the CPS enabled setting results in 5.9% lower average RTT and 14.7% higher average throughput. In addition, the CPS enabled setting reduces the RTT variance caused by TDD policy changes.

Comparison with RL schemes. While we employ the Soft-Actor-Critic (SAC) algorithm to train Wixor's RL agent, a variety of algorithms can be utilized within the abstract RL framework described in §4.4.2. Here, we compare Wixor with Deep Q-Network (DQN) and Proximal Policy Optimization (PPO). DQN is a "tabular" q-learning method that represents the RL policy as a table with discrete entries for all state-action pairs, whereas PPO is a recent policy gradient method. We train PPO in the same way as Wixor while DQN uses fine-grained state and action space quantization. Table 4.9 presents the average QoS reward r_t (Eqn. 4.4.2) attained by each method on the test traces. The results indicate a substantial performance disparity (27.9%) between the

Table 4.10: Varying number of neurons and filters (1D-CNN unit).

# Neurons / Filters	Avg. Test Reward r_t
32	0.82 ± 0.28
64	0.93 ± 0.22
128	0.94 ± 0.18
256	0.94 ± 0.14

Table 4.11: Varying number of hidden layers.

# Hidden Layers	Avg. Test Reward r_t
1	0.93 ± 0.22
2	0.93 ± 0.28
3	0.89 ± 0.26
6	0.81 ± 0.34

tabular scheme and Wixor, underscoring the inadequacy of tabular RL schemes in capturing the complexities of the RAN environment. Conversely, PPO demonstrates performance comparable to Wixor's SAC method, with only a 3.2% gap.

NN architecture. We conduct a parameter sweep to evaluate the impact of various NN parameters on r_t. Tables 4.10 and 4.11 present the average test reward corresponding to different numbers of neurons and hidden layers, respectively. Our findings indicate that performance plateaus once the number of filters in the 1D-CNN and the number of neurons each exceed 64. Additionally, the results reveal that the NN with a single hidden layer yields the best performance.

Training time. We quantify the overhead associated with training Wixor's RL agent. The training process encompassed approximately 300,000 iterations, equivalent to 3.5 hours of runtime. Each iteration required 42 milliseconds and involved concurrent parameter updates for 8 agents. It is important to note that this overhead represents a one-time, offline computational cost.

4.8 Discussion & Conclusion

Summary. We presented Wixor, a practical system for dynamic TDD policy adaptation in 5G/xG radio access networks. Wixor decomposes the problem into two stages: a proactive demand customization engine that forecasts future UL/DL slot percentages using compact, transferable

BS-level features and a context-aware policy provision stage that (i) smooths predictions conservatively to avoid destabilizing transport- and application-layer adaptation, and (ii) derives a QoS-aware TDD pattern by balancing inter-slot delay against guard-period overhead. We implemented Wixor on an srsRAN-based prototype, trained its RL agent in a trace-driven simulator, and evaluated the design using extensive ns-3 simulations and over-the-air experiments with real application workloads. Across a wide mix of applications (edge video analytics, vehicle perception, live ingest, VoD, conferencing, and file transfer), Wixor consistently improves QoS/QoE metrics relative to static and prior dynamic baselines while remaining lightweight and standards-compliant.

Limitations and future work. We acknowledge several limitations in the current work and identify promising directions for extension:

1. **Inter-BS interference and coordination.** Wixor currently operates using BS-local features and does not explicitly account for cross-link interference or coordinated multi-BS policies. Extending Wixor to incorporate multi-BS context (e.g., neighboring BS load, interference maps, or a RIC-style controller) could enable coordinated TDD adjustments that further improve network-wide performance but would require redesigning parts of the reward function and the training pipeline to account for multi-agent interactions.

2. **mmWave-specific considerations.** Our evaluation focuses on mid-band 5G (the most common TDD deployment for private and many public networks). While the core ideas behind Wixor apply to mmWave, mmWave introduces additional considerations (e.g., more frequent beam management, very small cell footprints, rapid blockage) that warrant specialized adaptation of the prediction and policy-derivation modules.

3. **Online RL and continual learning.** For safety and stability we trained Wixor's RL agent offline in a trace-driven simulator and transferred the learned model to the live BS; runtime training/exploration was intentionally avoided to prevent harming user QoE. A next step is a safe continual-learning framework that incrementally refines the model online (e.g., via off-policy updates, constrained exploration, or human-in-the-loop validation) while strictly bounding negative impacts during exploration.

4. **Dependence on 5QI for ρ_t.** The default mechanism to compute buffering tolerance ρ_t relies on 5QI markings to identify latency-sensitive flows. This is practical for private deployments where bearer configuration is under operator control, but public deployments or legacy UEs may lack reliable 5QI tagging. Practical fallbacks include operator-configured defaults, heuristic estimates from observed buffer/HOL delay distributions, or lightweight application-probing techniques.

5. **Broader RAN ecosystem integration.** Wixor is implemented as a modular BS-side service and is compatible with current 3GPP signaling. Integrating Wixor with emerging software-defined RAN paradigms — for example, OpenRAN's RIC or network-slicing orchestrators — is an exciting direction that would enable multi-tenant policy awareness and per-slice TDD optimization.

Practical considerations. Wixor was built to be deployable with minimal changes to existing infrastructure: it is 3GPP-compliant, implemented atop a standard-compliant srsRAN stack, and uses inputs (CQI, BSRs, RLC/MAC counters) that are available at typical BS deployments. Its computational footprint is modest: inference runs within milliseconds on commodity server CPUs

(and faster with GPUs), and the offline RL training is a bounded, one-time cost. The conservative policy-smoothing module reduces the risk of destabilizing transport/application adaptation, helping ensure real-world robustness.

Takeaways. Our measurement study highlighted that current deployments tend to use static, DL-biased TDD patterns that fail to adapt to rapid and frequent fluctuations in UL/DL demand; naive reactive adjustments can hurt QoE due to the highly dynamic radio environment and transport-layer interactions; and the arrangement of UL/DL symbols (inter-slot delay) matters substantially for latency-sensitive workloads. Wixor addresses these issues via a practical combination of predictive RL-based demand estimation, compact BS-level features for transferability and scalability, and a QoS-aware policy derivation that explicitly handles guard periods and inter-slot delay. The result is a system that improves application performance across diverse workloads while remaining implementable on real BS platforms.

Closing. Dynamic TDD is a powerful tool for future 5G/xG networks, but realizing its benefits in practice requires careful attention to measurement-driven prediction, protocol constraints (guard periods, symbol arrangement), and interactions with transport/application logic. Wixor takes a concrete step toward that goal by combining a pragmatic, standards-compatible implementation with demonstrated gains in both simulation and over-the-air experiments. We release our implementation and measurement artifacts to facilitate further research and operator trials, and we hope this work encourages continued investigation into coordinated, safe, and application-aware TDD adaptation for next-generation RANs.

Chapter 5

Performance-Driven Connectivity Management in 5G

5.1 Introduction

5G New Radio (NR) spans a wide range of frequency bands — Low-Band (<1GHz), Mid-Band (1–6GHz), and High-Band (mmWave, 24–40GHz). Since its rapid deployment beginning in 2019, 5G has largely coexisted with 4G/LTE: many operators run Non-Standalone (NSA) with Dual Connectivity (DC) while others offer Standalone (SA). These mixed architectures, together with Carrier Aggregation (CA) used to boost bandwidth and throughput [45, 189], produce highly heterogeneous multi-cell environments in which a single base station may expose multiple cells across bands and technologies.

Despite significant research on resource allocation within a cell [41, 71, 78, 69, 167], the foundational problem of *which cell(s) should serve a given UE* — i.e., how to choose the cell combination for selection, reselection, handover, and CA/DC — remains under-explored. 3GPP specifies the procedures for these actions (collectively referred to here as connectivity management, CM), but it leaves many of the policy and objective choices to operators. As networks become more heterogeneous, this operator flexibility increasingly matters for user-level performance.

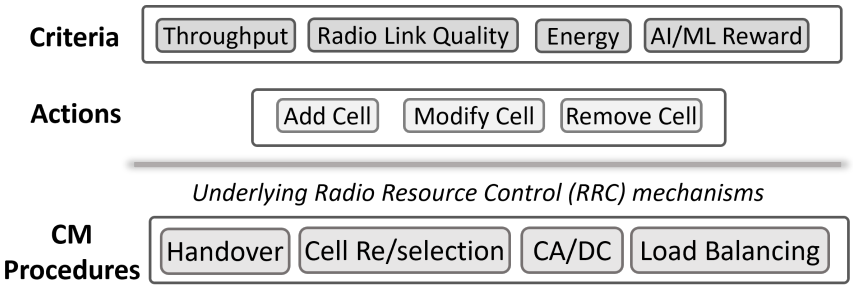


Figure 5.1: Decoupling CM criteria and underlying CM procedures via an abstraction layer.

Issues with legacy CM. Legacy CM schemes have three important limitations that hinder end-user performance and operational flexibility. **(i)** Legacy CM is largely *performance-oblivious*: decisions are made primarily on radio link quality (e.g., RSRP/RSRQ) without directly accounting for throughput, latency, energy, or other higher-level objectives. **(ii)** CM procedures are treated independently (cell selection, handover, CA/DC, etc.), which leads to redundant logic, complex management, frequent misconfigurations, and — in the worst cases — outages and instability seen in operational networks [50, 91, 192, 135, 139, 54, 140]. **(iii)** CM is network-centric and uniform across UEs, but different flows and devices have different requirements: a latency-sensitive flow may prefer a low-latency cell while a backlog-heavy transfer may prefer high aggregate throughput. Existing schemes do not support this per-UE/per-flow personalization [49, 89].

These limitations help explain why some observers consider 5G underwhelming for real applications even though the technology can deliver much higher raw capacity [168, 141].

Our proposal. We introduce a unified, performance-driven connectivity management framework, OPCM, that decouples *what* performance objectives the network desires (throughput, latency, energy, fairness, etc.) from *how* the CM machinery (selection/reselection/handovers/CA/DC) is exercised to achieve them. Concretely, OPCM provides a higher-level abstraction layer that

accepts pluggable performance criteria and maps them to CM actions while remaining backward-compatible with existing 3GPP procedures (see Figure 5.1). This separation lets operators and applications express diverse objectives without changing the low-level CM protocols.

Under OPCM, the network can opportunistically select cell combinations to match objectives: e.g., aggregate high-band cells for throughput-hungry transfers, pick low-latency cells for real-time flows (higher numerology/subcarrier spacing), or favor energy-efficient cells for lightweight IoT uplinks. When coverage is the priority, traditional link-quality rules remain valid.

Why this is timely. Three empirical observations motivate OPCM: **First,** our measurements across 12 cities in five countries show *wide availability and stable heterogeneity* of cell deployments (§??). In the median case a UE has access to 7+ distinct cell combinations spanning NSA-5G, SA-5G, and LTE across multiple bands — a rich decision space that legacy CM leaves mostly unused. **Second,** the diversity of accessible cells hides *large untapped performance gains.* In large-scale experiments we observe improvements up to 70.1% on QoE metrics (video bitrate, per-frame latency, energy consumption) when opportunistically selecting better cell combinations (§??). **Third,** recent 3GPP releases introduce more flexible CM primitives (e.g., conditional and DAPS handovers [125]), lowering the bar for deploying smarter CM logic and making a performance-driven approach more practical.

Challenges. Adopting a performance-driven CM framework raises several non-trivial challenges: *(i)* scalability — making coordinated CM decisions for tens to hundreds of UEs with heterogeneous objectives; *(ii)* policy compliance — satisfying operator constraints and spectrum management policies; *(iii)* accurate yet lightweight profiling — estimating the expected performance of candidate cell combinations without intrusive probing that would degrade service; and

Table 5.1: Comparison of existing CM techniques with OPCM.

Features		Legacy	iCellSpeed	OPCM
Supports Perf. Metric	**Link Quality**	✓	✓	✓
	Throughput	✗	✓	✓
	Latency	✗	✗	✓
	Energy	✗	✗	✓
Fair to other UEs		✓	✗	✓
Respects RAN Policy		✓	✗	✓
Is 3GPP Compliant		✓	✗	✓

(iv) execution fidelity — applying CM changes (handovers, CA reconfigurations) without causing harmful packet interruptions.

Opportunistic Performance-driven CM. We present OPCM to tackle these challenges. It operates opportunistically, ensuring at least the performance of legacy CM while seeking additional gains. A key design question is: *who* should be responsible for performance-driven CM? A user-side solution, like iCellSpeed [49], is immediately deployable on commercial 4G/5G networks; however, since each UE would individually handle CM, it cannot guarantee cross-user fairness or fulfill operator-imposed polices. In contrast, a centralized network-side solution can coordinate multiple UEs, enforce operator policies at various scopes, and facilitate fairness. OPCM sits on the base station, requiring no UE-side modifications. It remains 3GPP-compliant and fully backward-compatible with legacy CM. OPCM also exposes a modular custom metric registration API that reduces CM configuration complexity. Table 5.1 summarizes key differences between CM schemes, highlighting OPCM's broader metric support, policy compliance, and fairness guarantees.

Core ideas and components. OPCM is built from three interacting components:

- **Smart Decision Framework (§5.4)** — reduces the multi-UE combinatorial problem to tractable single-UE subproblems by pruning cell combinations that would violate fairness

or operator policies, then opportunistically selects promising candidates from the pruned set.

- **Hybrid Profiling Engine (§5.5)** — combines passive estimation and selective lightweight probing, exploiting cross-correlations among cell combinations to infer performance with minimal overhead.

- **Robust Execution Module (§5.6)** — applies queuing-aware delayed reconfiguration and safe fallbacks (reverting to legacy link-quality CM when gains are marginal) to reduce service interruptions during reconfiguration.

Prototype and evaluation. We implement OPCM on a programmable over-the-air testbed using open-source LTE/5G stacks [158, 156], totaling 6.1K+ lines of code. Our evaluation combines small-scale over-the-air experiments (realism) with large-scale trace-driven simulations (generalizability), multiple handset models, various mobility patterns (walking, driving, campus, suburban), and real application workloads (VoD, video analytics). Key results include: **(i)** OPCM yields up to 65.2% and 28.1% higher average QoE than legacy CM and a UE-side CM solution (iCellSpeed) respectively (§5.8.2); **(ii)** under mobility OPCM matches legacy CM in stability while finding higher-performing cell combinations in the wild (§5.8.3); and **(iii)** OPCM respects operator policies — cross-UE fairness remains within 98–99% of legacy CM, spectral efficiency improves by 2–3%, and system overhead is modest (3.1–6.5%) while supporting advanced CA/DC settings (§5.8.4).

5.2 Measurement and Motivation

5.2.1 Measurement Setup

Operator, Band, and Technology. We study three major commercial cellular operators (T-Mobile, AT&T, and Verizon) in the United States. These operators deploy services using LTE, NSA-5G, and SA-5G across Low-Band, Mid-Band, and mmWave frequencies. To extend our analysis geographically, we also collect measurements in Europe using local operators (Vodafone, Telekom, SFR, and Orange). European operators support LTE and NSA-5G in Low-Band and Mid-Band ranges [81], while only Vodafone offered SA-5G during our study period. Because mmWave coverage remains sparse in both the US and Europe [174, 58, 81], our analysis primarily focuses on Low-Band and Mid-Band deployments (600–3300 MHz), which constitute the dominant coverage layers in current networks.

Measurement Applications. We develop an Android measurement application to evaluate different workloads in the wild.

(i) Live Video Streaming. Our streaming pipeline consists of an RTSP media server [151], an ffmpeg-based encoder [62], and a lightweight Android RTSP client [51]. We transmit pre-recorded video at 30 FPS with a target bitrate of 34 Mbps, representative of 4K streaming requirements.

(ii) Video Conferencing. We implement a minimal Android WebRTC client using a commercial SDK [104], ensuring compatibility with standard real-time communication protocols.

(iii) Energy Profiling. We estimate UE energy consumption using model-based profiling. Following Narayanan et al. [122], we collect current-draw traces using a Monsoon Power Monitor [115] at different throughput levels and fit linear regression models to derive device-specific

energy models. This avoids reliance on deprecated Android APIs or external tethered measurement setups during in-the-wild experiments.

All applications are hosted on a university server with 4 Gbps+ bandwidth, ensuring that the Internet does not become a bottleneck. We additionally collect geolocation, mobility speed, signal strength, base-station identifiers, and ping latency using Android APIs.

Methodology. We use multiple smartphone models—Samsung Galaxy S10 (S10), S20 Ultra (S20U), S21 Ultra (S21U), and S22+ (S22+)—to minimize device-specific bias. Devices are tethered to a laptop via USB3 and synchronized using ADB commands [46]. To extract lower-layer signaling from unrooted UEs, we use Accuver XCAL [18]. To reduce interference across devices, we lock UEs to separate BSs whenever possible, benchmark devices prior to experiments, and repeat each experiment multiple times for consistency.

5.2.2 Wide Availability of Cell Deployments

To map the footprint of available cells, we use band-locking tools (∗#2263#) to enumerate cells accessible to each UE. Connectivity is verified using ping. We conduct experiments across 90+ locations spanning 12 cities in the US and Europe, repeating each experiment three times per location.

Our analysis reveals that *cell deployments are both abundant and spatially stable.* Figure 5.2 shows the number of unique base stations visible at each location. At any location, more than three BSs are accessible approximately 94% of the time. The median case observes 3–6 BSs, with some locations reaching up to eight. When CA/DC is considered, the number of unique cell combinations exceeds seven in the median case.

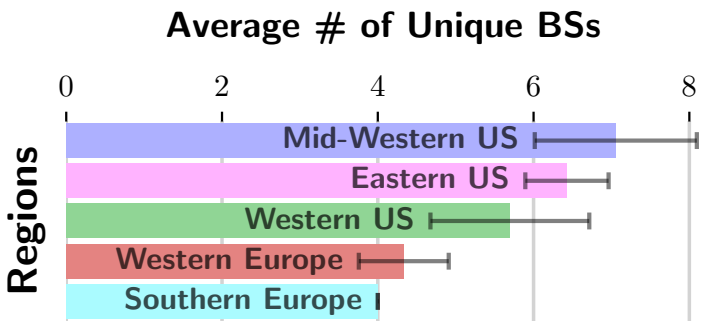


Figure 5.2: Density of cell deployments across different regions.

We observe denser deployments in the US compared to Europe. This disparity arises from the limited availability of SA-5G and NSA-5G Low-Band in Europe during our study period, whereas US operators widely deployed NSA-5G Low-Band [72, 81].

To evaluate *temporal stability*, we conduct a 13-month measurement campaign across four fixed locations (campus, residential, downtown, and airport). Across all sites, 5–6 cell combinations remain available 95% of the time. Changes only occur when operators introduce new bands (e.g., SA-5G midband or NSA C-band), confirming that cell availability is stable over long timescales.

5.2.3 Performance Diversity in Cell Deployments

With the widespread availability of multiple cell combinations, an important question arises: do these combinations exhibit meaningful performance diversity across metrics such as throughput, latency, and energy efficiency? If so, how large is the gap between legacy CM—optimized primarily for connectivity—and performance-driven selection?

Case study. We evaluate performance diversity using a 740 m $\times$ 510 m loop on our university campus. Using XCAL, we confirm that a T-Mobile UE can consistently access four distinct PCells (and corresponding PSCells). Four S22+ UEs are locked to these four cell combinations, while a

Table 5.2: Cell combinations used in our experiments. The number of carriers (SCells) may vary over time.

Label	Cell Combination
S1	113^4(1955)/39^5(626)
S2	107^4(1935)/278^5(2510)
S3	59^4(2145)
S4	46^5(636)
S5	Legacy CM

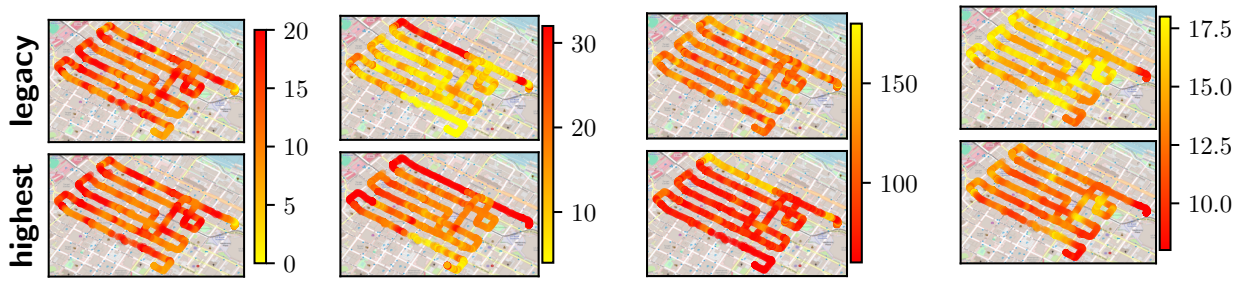
fifth UE uses legacy CM. The combinations span LTE, NSA-5G, and SA-5G configurations, with CA enabled. Table 5.2 lists the configurations (S1–S5). Devices are placed side-by-side and collect measurements concurrently while walking.

Each UE runs three applications: live video streaming, video conferencing, and energy profiling (§5.2.1). For streaming, rebuffering is negligible (<0.03%), so we report video bitrate. For conferencing, we report per-frame latency. We also record SINR for PCells/PSCells.

Figure 5.3 shows SINR, video bitrate, conferencing latency, and energy consumption. Legacy CM achieves comparable SINR to the best configuration, confirming its focus on connectivity. However, no single cell combination performs best across all metrics. The performance gap between legacy CM and the best configuration ranges from 27.0% to 70.1% across throughput, latency, and energy efficiency. This diversity arises from factors such as operator deployment strategies, channel variability, device capabilities, and CA configuration priorities.

Furthermore, even for a single metric, no configuration consistently dominates. For example, one configuration achieves the highest bitrate 43.2% of the time but offers the lowest latency only 11.3% of the time. This indicates that optimal cell choice is workload-dependent.

Large-scale throughput characterization. We extend the analysis using large-scale experiments across mobility scenarios including walking, driving, public transit, indoor, and stationary



(a) SINR (dB) (b) Video bitrate (Mbps) (c) Per-frame latency (ms) (d) Energy cons. (mW/Mbps)

Figure 5.3: A case study to quantify the performance gap between legacy CM and the highest performing cell combinations. The color gradient ranges from red (highest performance) to yellow (lowest performance).

conditions. UEs are band-locked to representative LTE, NSA, and SA configurations. Each UE runs uplink and downlink file transfers.

Figure 5.4 shows the throughput CDF comparing legacy CM against the highest-performing cell combination. Although legacy CM provides the best link quality 83.8% of the time, it achieves the highest downlink throughput only 16.6% of the time and the highest uplink throughput only 3.6% of the time. The median throughput gap between the best combination and legacy CM is 143.1% (62 Mbps) for downlink and 86.8% (9 Mbps) for uplink. In the median case, two to three alternative cell combinations outperform the legacy selection.

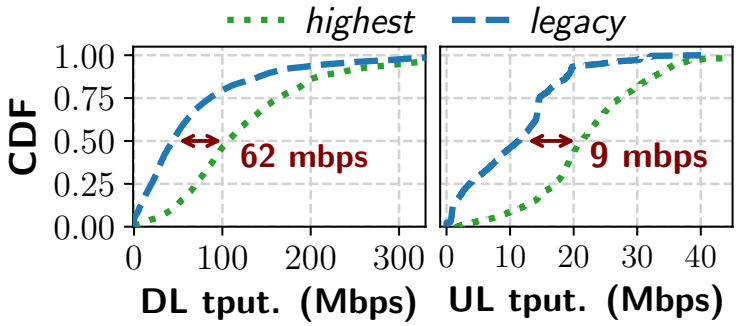


Figure 5.4: Performance breakdown of cell combinations in our dataset.

Importantly, uplink and downlink optimal combinations coincide only 42.1% of the time, suggesting that CM decisions should be decoupled across directions.

Key takeaway. These findings demonstrate that *no single cell combination consistently optimizes all performance metrics.* Instead, heterogeneous deployments create substantial, untapped opportunities for performance improvement through adaptive, performance-driven connectivity management.

5.3 Opportunistic Performance-driven Connectivity Management (OPCM)

Here, we introduce OPCM (**O**pportunistic **P**erformance-driven **CM**) framework. Its design needs to overcome several challenges.

[C1] Performing CM for 10s–100s of UEs presents significant complexity due to varying performance requirements and heterogeneous cell deployments. **[C2]** The framework must adhere to operator policies, such as ensuring cross-user fairness and balancing cell loads. **[C3]** Network performance profiling entails either actively switching to cells with potentially degraded performance or using passive approximation techniques, which are often error-prone. The challenge lies in determining the best approach to minimize overhead while ensuring accuracy. **[C4]** CM procedures, such as handovers, can cause data interruptions. Minimizing these disruptions is critical to maintaining seamless user experiences.

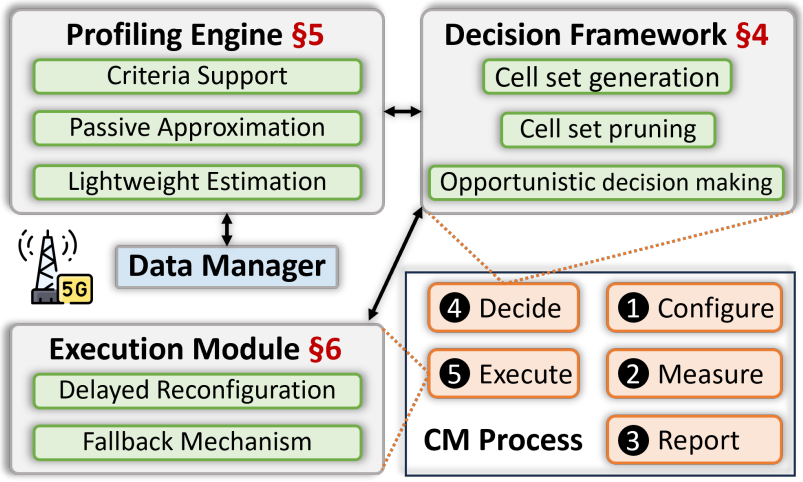


Figure 5.5: The overall design workflow of OPCM.

5.3.1 Design Overview

Figure 5.5 illustrates how OPCM works compared to legacy CM schemes. Typically, the serving BS configures ❶ a UE to continuously measure ❷ neighboring cells based on radio link quality criteria. Once a configured criterion is met, the UE reports ❸ it to the serving BS. The serving BS then decides ❹ whether to initiate a CM procedure. If yes, the serving BS sends a command to the UE and executes ❺ the procedure.

In this workflow, ensuring that OPCM effectively selects the best serving cells according to configured performance criteria — without degrading other UEs' performance — is central to its adoption. Therefore, OPCM is designed to be *opportunistic*: it guarantees performance at least as good as legacy CM while searching for additional gains.

OPCM replaces only the decision (❹) and execution (❺) stages with its own *Decision Framework* and *Execution Module* modules, respectively. The underlying 3GPP CM procedures remain unchanged. It introduces *Profiling Engine* to support diverse performance criteria, and *Data Manager* to collect network state, signal quality, and measurement reports readily available at the BS.

Performance-driven CM requires solving a joint optimization problem across UEs, cell deployments, performance metrics, and RAN policies, leading to scalability challenges (**C1**).

To address this, the **Smart Decision Framework (§5.4)** decomposes the monolithic multi-UE CM problem into multiple single-UE CM decisions. It further reduces complexity by eliminating infeasible cell combinations. OPCM first constructs a candidate cell set containing all feasible cell combinations. Based on empirical measurements from public networks, we significantly reduce the search space of possible combinations. The system then performs cell-set pruning to remove combinations that violate operator policies (**C2**), further shrinking the candidate set. Finally, OPCM opportunistically selects a cell combination using performance estimates from *Profiling Engine*, balancing the tradeoff between exploring new combinations (profiling cost) and exploiting known high-performing ones (**C3**).

The **Hybrid Profiling Engine (§5.5)** estimates the performance of cell combinations using criteria such as radio link quality, throughput, latency, and energy efficiency. OPCM leverages cross-correlation among combinations to opportunistically approximate performance without actively switching UEs whenever possible (**C3**).

The **Robust Execution Module (§5.6)** carries out CM decisions. It employs delayed reconfiguration to reduce data interruptions (**C4**) and includes a fallback mechanism that reverts to legacy link-quality-based CM if expected gains are marginal.

Deployment Practicality. *(i)* OPCM operates as a lightweight BS-side service and requires no UE modifications. *(ii)* It adheres to standard 3GPP mechanisms and does not replace UE-side RRM measurements required for access and channel adaptation. *(iii)* OPCM is backward-compatible: we implement legacy radio link-quality-based CM within it for benchmarking (§5.8.3). *(iv)* Similar to X2/Xn handovers, it supports BS coordination over X2/Xn interfaces for shared

profiling and coordinated CM decisions (§5.9). *(v)* To preserve opportunism (never worse than legacy), OPCM includes hysteresis, legacy fallback, and marginal-gain avoidance mechanisms. *(vi)* Since optimal uplink and downlink cell combinations may differ (§5.2.3), OPCM applies CM logic independently per direction. For brevity, we describe the downlink procedure in the remainder of this chapter.

5.4 Smart Decision Framework

5.4.1 Cell Set Generation

At any moment, each UE is under the coverage of multiple cells with varying configurations. The serving BS configures all connected UEs to report measurements of neighboring cells (❸ in Figure 5.5). Additionally, UEs can be configured with multiple serving cells, referred to as a cell combination in this paper (§2.4). OPCM leverages these radio link quality measurement reports, along with the UE's current cell combination, to construct a cell set $\mathcal{C} = \{c_i \mid i \in \mathbb{N}\}$, where each cell combination c_i can include multiple serving cells. Furthermore, OPCM continuously updates $\mathcal{C}$ during runtime whenever a new c_i appears or an old one disappears.

UEs are typically surrounded by 3–6 BSs (§5.2.2). Each BS can host multiple PCells and several SCells, which in the worst case can lead to a combinatorial explosion of possible cell combinations. Profiling all these combinations is prohibitively expensive.

Our measurements in Figure 5.6 show that operators do not arbitrarily activate SCells during CA. Instead, they follow threshold-based CA policies [53]. To characterize this behavior, we conducted controlled stationary experiments under line-of-sight conditions. Using iperf3 to generate downlink traffic ranging from 1–100 Mbps and collecting lower-layer CA traces with XCAL, we

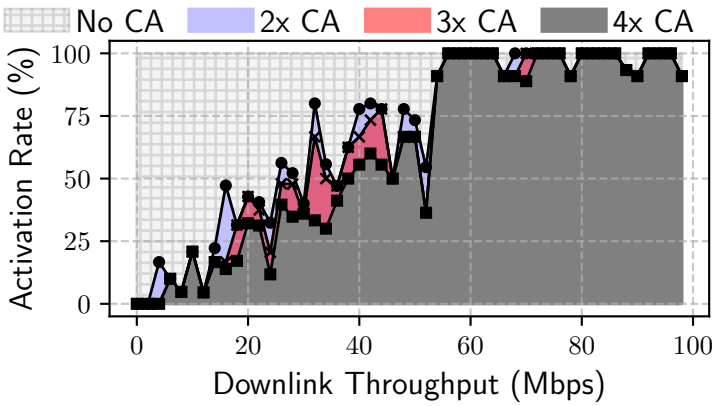


Figure 5.6: Group-based addition of carriers during CA.

observe two key findings. ***(i)*** Operators use relatively low traffic thresholds to activate additional carriers. For example, the second, third, and fourth carriers are triggered at approximately 4.4 Mbps, 7.8 Mbps, and 11.4 Mbps, respectively. ***(ii)*** More importantly, we frequently observe 4x CA even at modest sending rates, suggesting a *group-based* CA strategy in which multiple carriers are activated together rather than strictly sequentially.

These results indicate that, in practice, operators enable a limited and structured set of carrier combinations rather than exploring the full combinatorial space of possible SCell additions. Consequently, many theoretically feasible cell combinations never materialize in real deployments. Leveraging this empirical observation, OPCM restricts the candidate cell set $\mathcal{C}$ to combinations explicitly permitted by operator policies or those actually observed in measurement reports. This reduces the effective search space dramatically, to fewer than nine combinations in 95% of cases, without sacrificing practical coverage.

Table 5.3: OPCM RAN objective examples.

RAN Policy	$\mathcal{P}$	$\mathcal{P}^{max}$
User Fairness	$FI^t = \frac{(\sum_{u\in\mathcal{U}} \mathcal{T}_u^t)^2}{\lvert\mathcal{U}\rvert \sum_{u\in\mathcal{U}} (\mathcal{T}_u^t)^2}$	1.0
Band Load Balancing	$LDI^t = \frac{(\sum_{c\in\mathcal{C}} \mathcal{R}_c^t)^2}{\lvert\mathcal{C}\rvert \sum_{c\in\mathcal{C}} (\mathcal{R}_c^t)^2}$	1.0

5.4.2 RAN Policy Compliance via Cell Set Pruning

A BS-side vantage point enables OPCM to respect operator Radio Access Network (RAN) policies. In this subsection we describe how policies are encoded and enforced.

Defining RAN policies. OPCM encodes policies using a flexible δ-constraint notation. For any policy $\mathcal{P}$ and candidate cell combination c_i, $\delta_{c_i}^{\mathcal{P}}$ denotes the operator's tolerance for violating $\mathcal{P}$ when moving a UE to c_i. For example, $\delta_{c_i}^{FI}$ bounds how much the fairness index (FI) may drop when migrating a UE to c_i.

Operators can express policies scoped per-UE (e.g., a UE cannot access mmWave cells), per-cell (e.g., CA priorities), or per-BS (e.g., bandwidth caps), and they provide rules for computing policy metrics. Table 5.3 gives two example policies:

- **User Fairness:** computes the Jain fairness index of long-term average throughputs $\mathcal{T}_u^t$ across users $\mathcal{U}$:

$$FI^t = \frac{\left(\sum_{u\in\mathcal{U}} \mathcal{T}_u^t\right)^2}{\lvert\mathcal{U}\rvert \sum_{u\in\mathcal{U}} (\mathcal{T}_u^t)^2}.$$

- **Load Balancing:** computes a load distribution index (LDI) over candidate combinations using resource block usage $\mathcal{R}_c^t$:

$$LDI^t = \frac{\left(\sum_{c\in\mathcal{C}} \mathcal{R}_c^t\right)^2}{\lvert\mathcal{C}\rvert \sum_{c\in\mathcal{C}} (\mathcal{R}_c^t)^2}.$$

Realizing policies (pruning). Given $\delta^{\mathcal{P}}_{c_i}$, OPCM evaluates a "what-if" scenario for moving a UE to candidate c_i at the next timestep: it computes the resulting policy metric $\mathcal{P}^{t+1}_{c_i}$. If the deviation from the maximum allowed value violates the tolerance,

$$\mathcal{P}^{\max}_{c_i} - \mathcal{P}^{t+1}_{c_i} \geq \delta^{\mathcal{P}}_{c_i},$$

then c_i is pruned from the UE's candidate set $\mathcal{C}$. Pruned combinations are not considered in subsequent decision-making. This pruning enforces operator constraints (C2) and reduces the computation needed in the decision stage.

5.4.3 Opportunistic Decision Making

Once OPCM prunes $\mathcal{C}$ and obtains performance estimates from the *Profiling Engine* (§5.5) for each $c_i \in \mathcal{C}$, the *Decision Framework* module must decide whether to retain the current cell combination or switch to another. The core challenge is balancing *exploration* vs *exploitation* (**C3**): exploit the best-known combination now, or explore another to update estimates and possibly gain more later.

Balancing exploration and exploitation. We model the problem as a non-stationary multi-armed bandit: each arm corresponds to a cell combination and reward distributions may change over time. OPCM uses an epsilon-greedy strategy with exponential decay of ϵ: explore with probability ϵ, exploit with probability $1 - \epsilon$. Concretely, ϵ is decayed each decision epoch by a factor determined by λ, and empirical tuning shows good performance when $\lambda \approx |\mathcal{C}|/(2I)$, where I is an estimate of the typical cell-change interval (§5.8.5).

Gain-aware greedy exploitation. During exploitation, OPCM picks the candidate with the highest estimated performance $\mu^t_{c_i}$. However, switching entails reconfiguration cost and possible data-plane interruption (C4). Therefore, OPCM only switches to a new combination $\hat{c}^{t+1}$ if its estimated gain exceeds the current combination $\hat{c}^t$ by a hysteresis H:

$$\mu^t_{\hat{c}^{t+1}} > \mu^t_{\hat{c}^t} + H.$$

The hysteresis avoids oscillations and spurious switches; in our evaluations we use $H = \frac{1}{2}\sqrt{\mu^t_{\hat{c}^t}}$ unless otherwise specified.

Smart-random exploration. If exploration is chosen, OPCM does not pick uniformly at random. Instead, it computes an *exploration weight* $w^t_{c_i}$ for each c_i that balances (i) historical estimated performance $\mu^t_{c_i}$ and (ii) staleness measured by time since last use $lu^t_{c_i}$. We normalize both terms and combine them equally:

$$w^t_{c_i} = \frac{1}{2}\left(\frac{\mu^t_{c_i}}{\sum_j \mu^t_{c_j}}\right) + \frac{1}{2}\left(\frac{lu^t_{c_i}}{\sum_j lu^t_{c_j}}\right).$$

OPCM then samples a candidate according to the distribution induced by $w^t_{c_i}$. This prioritizes combinations that have reasonable historical performance and have not been probed recently, while avoiding repeatedly exploring poor performers. Newly observed combinations receive larger initial $w^t_{c_i}$, so they are prioritized for early exploration.

Practical notes. *(i)* Turning exploration off (set $\epsilon = 0$) recovers the legacy link-quality-based CM behavior, making OPCM's deployment incremental. *(ii)* The per-UE decision approach, combined with aggressive pruning and policy constraints, scales to tens–hundreds of UEs while still enabling coordinated fairness control at the BS.

5.5 Hybrid Profiling Engine

The *Profiling Engine* estimates the performance of candidate cell combinations according to the configured performance criterion. A central challenge is the absence of performance data for inactive cell combinations (**C3**). Since a UE can only operate on one cell combination at a time, directly measuring all combinations would incur prohibitive overhead.

To address this, the *Profiling Engine* introduces a passive performance approximation technique that leverages cross-correlation among cell combinations. This approach enhances estimates obtained via exploration while minimizing active switching overhead. Ultimately, the *Profiling Engine* produces the estimated performance metric $\mu^t_{c_i}$ for each $c_i \in \mathcal{C}$.

To compute $\mu^t_{c_i}$, the *Data Manager* collects the requisite raw data (e.g., the measured performance metric $m^t_{\hat{c}}$ of the active cell combination $\hat{c}$ at time t).

5.5.1 Passive Performance Approximation

Our measurements indicate that the performance of certain cell combinations is often correlated. Historical correlation patterns can therefore help infer the performance of inactive combinations.

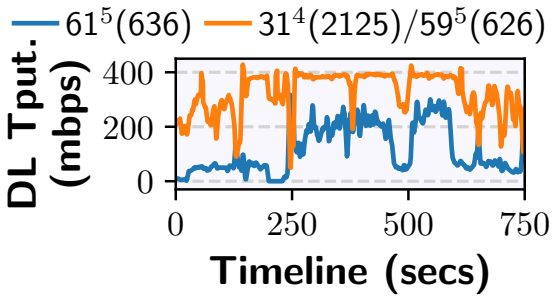


Figure 5.7: Example of correlation between two cell combinations.

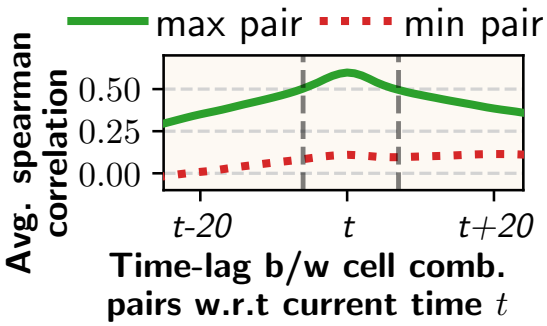


Figure 5.8: Time-Lagged Cross-Correlation (TLCC) across cell combinations.

Prior works [90, 93, 172, 30] have leveraged cross-band link-quality estimation at the physical layer. In contrast, we identify an opportunity at the network-performance level: multiple cell combinations frequently exhibit correlated throughput, latency, and energy behavior.

Figure 5.7 shows a representative trace illustrating synchronization between two cell combinations. Performance troughs and stable regions align closely. We attribute this to three factors: **(i)** shared cells or component carriers across combinations, **(ii)** common environmental events such as blockage during mobility, and **(iii)** shared BS-side congestion affecting resource scheduling.

To quantify this systematically, we use the large-scale dataset from §5.2.3 and compute Time Lagged Cross-Correlation (TLCC) [150]. TLCC accounts for the fact that only one cell combination can be active at any time, introducing measurement lag between combinations.

For two cell combinations a (active) and b (inactive), we compute:

$$Corr^{t}_{a,b} = \max_{\delta t} Corr(a(t - \delta t), b(t))$$

where δt is the lag that maximizes correlation. We use a 2-minute sliding window to compute TLCC.

Figure 5.8 shows that many combinations exhibit non-trivial correlation (often > 0.5) within a short lag window (typically < 5 seconds). This motivates leveraging recent measurements of the active combination to improve estimates for correlated inactive combinations.

We update inactive combination estimates using:

$$m_{c_i}^t = \mu_{c_i}^t \left[1 + Corr_{\hat{c},c_i}^t \cdot \frac{m_{\hat{c}}^t - m_{\hat{c}}^{t-1}}{m_{\hat{c}}^{t-1}} \right] \tag{5.1}$$

Intuitively, this equation applies the measured gradient of the active combination $\hat{c}$ to the inactive combination c_i, weighted by the correlation strength $Corr_{\hat{c},c_i}^t$. If $Corr_{\hat{c},c_i}^t > 0.5$ and the lag is small (< 5 seconds), we apply this approximation. Otherwise, we retain the most recent estimate for c_i to avoid inaccurate predictions under weak correlation (*e.g.,* high mobility or different BSs). We observe similar correlation behavior for latency and energy metrics.

5.5.2 OPCM Performance Criteria

Lightweight Performance Estimation. Unlike application-layer systems that require precise network performance forecasts, OPCM only requires comparative ranking of cell combinations. Since the decision stage performs an $\arg\max$ over $\mu_{c_i}^t$, ordinal accuracy is more important than absolute precision.

We therefore adopt a lightweight Exponential Weighted Moving Average (EWMA) filter to update performance estimates:

$$\mu_{c_i}^t = \alpha \cdot m_{c_i}^t + (1 - \alpha) \cdot \mu_{c_i}^{t-1} \tag{5.2}$$

Trace-driven simulations show that $\alpha \in [0.5, 0.8]$ performs well (§5.8.3). More sophisticated predictors are left for future work.

Supported Performance Criteria. OPCM supports multiple performance metrics that are transparent to applications and computable at the BS. By default, we use 5G QoS Identifier (5QI) [**<empty citation>**] to infer the performance objective of each flow. Currently supported criteria include:

- **Radio link quality:** Directly uses UE-reported measurements.
- **Network throughput:** Derived from Radio Link Control (RLC) throughput measurements.
- **Network latency:** Approximated using RLC queue's head-of-line delay.
- **UE energy efficiency:** Estimated via offline energy models or preference lists built from power measurements (e.g., Monsoon or Android ODPM [22, 68]). Energy models predict consumption based on throughput, cell combination, and device type [122].

This modular criterion support allows OPCM to flexibly optimize connectivity decisions for diverse workloads without requiring application-side modifications.

5.6 Robust Execution Module

Once the *Execution Module* module receives the CM decision ($\hat{c}^{t+1}$) from the *Decision Framework* module, it first checks the UE's Radio Resource Control (RRC) state [136] and its current cell combination ($\hat{c}^{t}$). Based on these conditions, OPCM triggers the appropriate CM procedure — cell reselection, handover, or CA/DC reconfiguration — to transition the UE to the target cell combination. The execution command is delivered via a standard RRC reconfiguration message to the UE, preserving 3GPP compliance.

To improve robustness and minimize performance degradation during CM transitions, OPCM incorporates two additional mechanisms.

(i) Delayed Reconfiguration. CM procedures inevitably incur data-plane interruptions (**C4**). A closer inspection shows that interruption impact is significantly amplified when uplink or downlink transmission queues contain pending data.* This observation suggests an optimization opportunity: instead of immediately executing the CM decision, OPCM can delay the reconfiguration to allow transmission queues to drain, thereby reducing disruption.

To implement this, OPCM initializes a countdown timer with duration ρ upon deciding to switch cell combinations. During this interval, it continuously monitors uplink and downlink queues. The downlink queue is directly observable at the BS, while uplink queue status is obtained through periodic Buffer Status Reports (BSRs) [14].

If both queues become empty before the timer expires, OPCM immediately executes the CM procedure. Otherwise, it temporarily increases the UE's scheduling priority in an exponential

*Uplink data resides in UE-side buffers, while downlink data resides in BS-side queues before radio resources are assigned. During CM execution, the BS does not schedule data transmission for the UE, resulting in temporary interruptions.

manner to accelerate queue draining. If the timer reaches zero, OPCM proceeds with execution regardless of queue status.

The choice of ρ introduces a tradeoff. A small ρ may not provide sufficient time for queue drainage, while a large ρ risks stale decision timing and reduced responsiveness. Empirically, we set ρ to the median observed CM-induced data-plane interruption (120 ms, see §5.2.3), which balances responsiveness and disruption mitigation. Because CM operations are relatively infrequent and ρ is short, the temporary priority boost has negligible impact on long-term fairness.

(ii) Fallback Mechanism. OPCM also includes a lightweight fallback mechanism to reduce unnecessary computation and energy consumption. When a UE is in idle state or experiences minimal data activity (e.g., estimated performance $\mu^t_{c_i} < 1$ Mbps), OPCM temporarily reverts to legacy radio link-quality-based CM and passively monitors traffic conditions.

This mechanism ensures that performance-driven CM is invoked only when meaningful gains are possible, reinforcing OPCM 's opportunistic design principle: it never performs worse than legacy CM while seeking additional improvements.

5.7 Implementation

OPCM Prototype. OPCM is built on top of srsRAN [158, 156], an open-source 4G/5G software-defined radio suite. We modified the cellular protocol stack (4G/5G Layer 2) in srsRAN to implement OPCM in over 6.1K lines of C/C++ code.

First, we added logging functionality to the RLC, MAC, and PHY layers to support performance data collection. We then developed a modular CM engine atop the RRC layer. This engine

abstracts the underlying CM procedures and introduces support for diverse performance criteria while remaining fully 3GPP-compliant.

The *Data Manager* collects logs (e.g., UE measurement reports and performance metrics) at configurable periodic intervals. These logs are forwarded to:

(i) the *Profiling Engine*, which determines the appropriate performance criterion (based on UE 5QI) and computes performance estimates, and

(ii) the *Decision Framework* module, which maintains the candidate cell set $\mathcal{C}$ and performs strategic CM decisions.

The system time step length is fixed at $\Delta t = 1$ second.

Finally, the *Execution Module* module receives CM decisions from *Decision Framework* and triggers the appropriate CM procedure. It observes the UE's current RRC state and compares the active cell combination with the target combination $\hat{c}^{t+1}$. Based on the differences, it determines which cells must be added, removed, or modified. The corresponding RRC reconfiguration message is then sent to the UE to initiate the CM procedure.

Since the BS can assign absolute priorities to cells (Table 2.3), OPCM leverages this 3GPP-defined feature to configure the desired cell combination at the UE. The *Execution Module* module also oversees the delayed reconfiguration logic and fallback mechanisms described in §5.6.

Custom Metric Registration. OPCM supports custom performance metrics through a lightweight C/C++ API. As illustrated in Figure 5.9, metrics are registered using `register_metric()` with a name, callback function, and invocation interval. The *Profiling Engine* invokes each callback with raw RLC/MAC/PHY statistics for all UE–cell combinations. Returned values are normalized and forwarded to *Decision Framework* for ranking and selection.

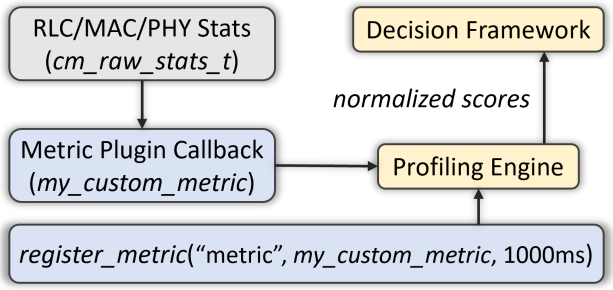


Figure 5.9: Custom metric registration flow in OPCM.

This modular interface enables extensibility without modifying core system logic and reduces the burden of manually configuring performance rules, directly addressing CM management complexity (§5.1).

Trace-driven Simulator. We developed a 4G/5G network simulator based on the *ns-3* LTE and NR codebase [126, 133]. The simulator mirrors the over-the-air prototype implementation of OPCM, including the CM engine, profiling, decision, and execution modules. We integrated trace-driven channel simulations and implemented a traffic generator for file transfer workloads. In total, we added or modified more than 4.9K lines of C/C++ and Python code.

5.8 Evaluation

We first build an in-lab end-to-end cellular network, given the lack of operator support and high cost of commercial BS deployment. Despite its limited scale, it provides a high physical-layer fidelity through real hardware and channel interaction. This is complemented by large-scale simulations to stress-test OPCM under a high density of users and cells, thereby ensuring reproducibility.

Figure 5.10: The over-the-air prototype testbed of OPCM.

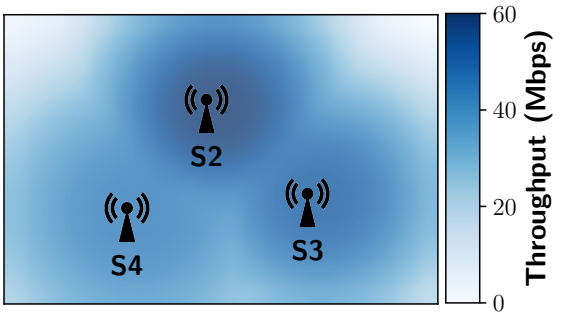


Figure 5.11: DL throughput density in 60m×40m test area.

5.8.1 Experimental Setup

(i) Over-the-air Testbed. The setup is shown in Figure 5.10. Each BS has two components: (i) a srsRAN-based eNodeB or gNodeB stack running on a laptop equipped with Intel Core i7 @ 3.00GHz CPU, and (ii) an RF frontend based on USRP B210 [171] software defined radio (SDR). The Open5GS Core Network (CN) [128] runs on a desktop machine. All apps are hosted locally with a 20 ms delay between the CN (packet gateway) and the remote server. We use ADB scripts to automate and time-synchronize experiments. All experiments are repeated at least 5×.

Experiment Settings. We set up three cell combinations on four B210s. NSA-5G needs two RF frontends, one for 4G and one for 5G. The frequencies for these cell combinations are according to **S2-4** in Table 5.2. Each one uses 20 MHz bandwidth with 64 QAM and 256 QAM MCS tables for uplink and downlink, respectively. All BSs use the proportional fair scheduler with default srsRAN parameters. We put the testbed (Figure 5.10) in an empty open parking garage to eliminate environmental noise. Figure 5.11 plots a density map showing maximum downlink throughput at any location. We freely walk at ∼3 mph during our experiments with UEs in hand.

Comparative Approaches. (i) The legacy approach uses the radio link quality-based CM to choose UE's cell combination. We use srsRAN's default criterion parameters. **(ii)** iCellSpeed [49]

is a UE-side solution to increase UE's network throughput. For a fair comparison, we implement a network-side version of iCellSpeed. We modify its *iCustomize* module since CM procedures can be directly triggered from the BS. The *iprofile* module is set up as described in the paper.

COTS & Virtual UEs. We use a Google Pixel 7 (PX7) smartphone and apply a programmable sim card to register it with CN. The PX7 phone lacks the ability to configure 5QI, therefore, we fix the 5QI in OPCM and test one application use case at a time. For simulation experiments, each app sets up its data bearers with the appropriate 5QI value. We also use 30 (10 for each BS) ZeroMQ srsUEs [157] with virtual radios. These virtual UEs utilize real-world network traces to generate network traffic and channel traces to model realistic channel conditions.

Network and Channel Traces. We collect these traces with NG-Scope [179] and post-process them to match our BSs' numerologies (e.g., cell bandwidth). We scale up/down traces to increase/decrease the BS load at the start of an experiment (call it $sload$). Note that the BS load can change during the experiment if a CM procedure transfers the UE from one cell to another. The average $sload$ is set to be 67% unless otherwise stated.

Additionally, we configure srsRAN to utilize real-world channel quality traces collected with XCAL. Our 14 hrs+ trace *corpus* is a 4-dimensional tensor (trace #,cell combination, channel, time), where each entry corresponds to a wideband Channel Quality Indicator (CQI) value. We chop these traces across time, with each trace spanning 350 secs. We randomly select 10 traces (for 10 virtual UEs) from the *corpus* for each BS. Since we have collected these traces in different mobility scenarios, the heterogeneity and randomization ensure that each BS has UEs with diverse channel conditions.

RAN Objectives. When OPCM is not running, the fairness index and load balancing index is in the range of 0.85–0.95 for our testbed (see Figure 5.18). Therefore, we set OPCM delta

tolerance (δ_b^O) for fairness and load balancing constraints to the higher limit 0.15 (i.e., $1 - 0.85 = 0.15$). Later, we investigate the impact of δ on OPCM performance (§5.8.5).

(ii) Trace-driven Simulations. We use *ns-3* [126, 133] to test OPCM with advanced 4G/5G settings and large number of users. The setup is identical to the over-the-air testbed, with a few exceptions. We increase the number of cell combinations to five by adding two new settings: **S1** in Table 5.2 and a 5G cell operating at 2155 MHz. Each BS can now aggregate up to 4 carriers, increasing the bandwidth from 20 MHz to 100 MHz. 150 UEs (30 for each BS) repeatedly download a 256 MB file from the remote server for 8 mins. Realistic channel conditions are still modeled with our *corpus* of traces.

5.8.2 OPCM QoE Improvement

To evaluate OPCM under our over-the-air testbed setup, we develop a suite of four mobile apps with *diverse* workloads.

(i) VoD Streaming. Our VoD streaming experiments use a dash.js [48] player to stream a 4 min video. We mainly test buffer-based BOLA [155] and rate-based [96] adaptive bitrate (ABR) algorithms due to their popularity. The video is encoded at 6 unique quality levels (0.8–6.8 Mbps average bitrate). Figure 5.12 plots the normalized bitrate and stall percentage for our experiments. The QoE improves in the top right direction as indicated by the arrow. The results show that, compared to the legacy, OPCM improves the average bitrate by 25.1%. Similarly, it reduces the average video stall percentage by 65.2%. iCellSpeed offers slightly (3.1%) higher bitrate than OPCM, but also has a 0.2% higher absolute stall rate. iCellSpeed performs well for downlink

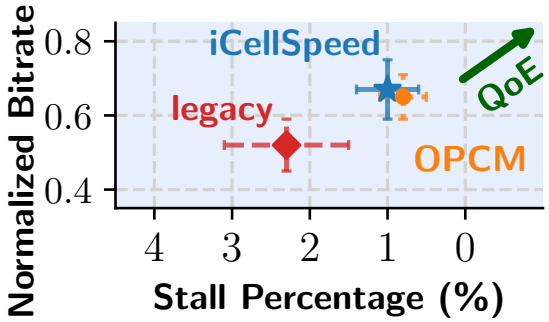


Figure 5.12: Comparing OPCM VoD streaming performance across baselines.

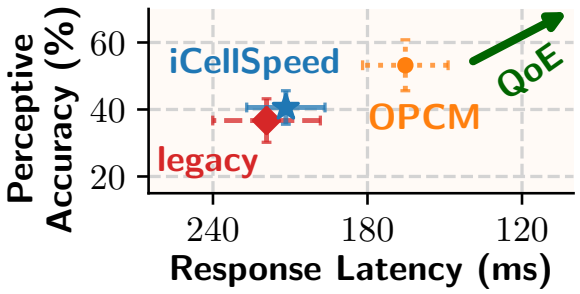


Figure 5.13: Comparing OPCM video analytics performance across baselines.

throughput-hungry applications but can lose performance for other application types and has a high memory footprint (details to follow).

(ii) Latency-critical Video Analytics. We select a popular video analytics task: Object detection (OD). OD app uses a state-of-the-art video analytics model (YOLOv4 [34]) deployed locally. Instead of sending camera feeds, both phones stream the same video frames from the COCO dataset [44] at 30 FPS. The perceptive accuracy (defined in [66, 88]) captures mean average precision for sending frames, and replaces a frame's inference with the last feedback if a response is not received within 200 ms. Figure 5.13 showcases that OPCM achieves 23.7% higher perceptive accuracy and 28.1% lower response latency than iCellSpeed on average. The performance difference can be attributed to two reasons: (i) iCellSpeed focuses on improving the throughput only while OPCM optimizes the performance criterion inferred from 5QI (see §5.5.2), and (ii) OPCM's queuing-aware delayed reconfiguration mechanism minimizes data-plane interruptions.

(iii) Uplink Video Ingest. We re-purpose Ant-Media's LiveVideoBroadcaster [105] to publish a pre-recorded video stream (1080p @ 30 FPS with 7.2 Mbps average bitrate). UEs send adaptive RTMP feeds [170] to a media server [23] deployed locally. We plot the sending bitrate and ingest delay for published video streams. Ingest delay, as defined in [193], is the time from when a

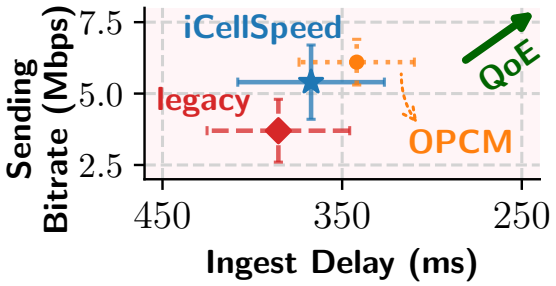


Figure 5.14: Comparing OPCM video ingest performance across baselines.

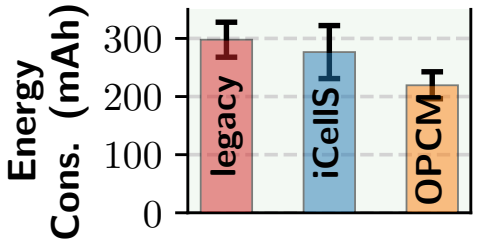


Figure 5.15: Comparing OPCM energy efficiency across baselines.

video frame is generated at the source to when its quality variants are available at the server for client download. Figure 5.14 depicts that OPCM achieves 12.9% higher average bitrate compared to iCellSpeed. Recall that uplink and downlink may have different highest performing cell combinations (see §5.2.3). OPCM can infer which direction to optimize unlike iCellSpeed that only optimizes the downlink throughput. Moreover, the uplink-intensive video ingest sees comparatively higher improvement than other downlink-heavy apps because the gap between legacy and the highest performing cell combination is wider for uplink (see §5.2.3).

(iv) Lightweight Fitness Tracker. We build an app to transmit live health data (e.g., heart rate) to the cloud. The app sends ∼100B messages every 0.25 ms. In addition, we configure OPCM to use energy consumption models and optimize PX7's energy efficiency (see §5.5.2). Figure 5.15 plots the overall energy consumed by the UE during a 30 mins experiment. It shows that OPCM improves the average energy efficiency by up to 26.3% and 20.6% over legacy and iCellSpeed, respectively. While 3GPP has not defined energy-specific 5QI values, Releases 16 and 17 introduce UE Assistance Information (UAI), enabling UEs to share energy-related preferences with the BS [4]. This, along with our evaluation, highlights the potential of energy-aware CM for applications like IoT.

5.8.3 OPCM Benchmarking

OPCM is backward-compatible, and performs as well as the legacy CM under mobility. To test if OPCM achieves its desired goals, we use radio link quality as the performance criterion inside OPCM and compare it with legacy CM. The goal is to see if OPCM triggers CM procedures exactly the same way as legacy CM does if the CM parameters (hysteresis, Time-to-Trigger, etc.) are same. We only run ping on the PX7 phone to keep the UE radio in active state. We split the parking garage area (Figure 5.11) into 6×4 lanes, mark the lanes with a tape, and walk on the tape vertically and horizontally to ensure reproducibility. We use WifiRttLocator [67] to position the UE with 1m accuracy. Overall, we collect 3 hrs+ of data with at least 65+ CM procedures triggered (mostly handovers and DC) for each setting (OPCM and legacy).

Given same mobility patterns, the CM procedures must be triggered at almost the same spot for both legacy CM and OPCM. Figure 5.16 plots the spread of the areas, where CM procedures are triggered repeatedly. To get this spread, we compute the convex hull for each spot where legacy CM procedures are triggered, and use that as a reference. We also calculate the overlap percentage of legacy's spread with OPCM's spread. A high mean overlap value of 89.2% shows that OPCM indeed works like legacy. We also conduct benchmarking simulation experiments, ensuring full reproducibility. The results reinforce our over-the-air testbed findings, i.e., OPCM triggers CM procedures at the same time and location as the legacy CM for radio link quality criterion.

OPCM's epsilon-greedy policy works effectively in the real world. To evaluate if OPCM's epsilon-greedy exploration can efficiently find the highest performing cell combination, we run a barebone version of our system on S22+ and test it under the same 740m×510m rectangular

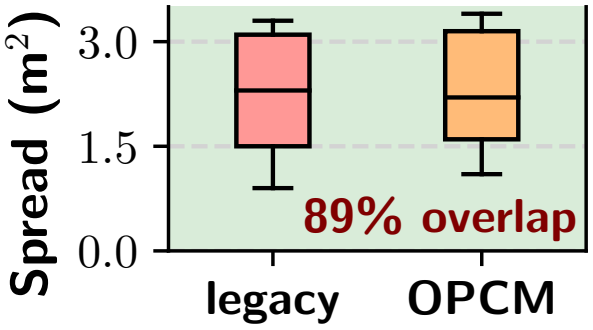


Figure 5.16: Benchmarking OPCM performance.

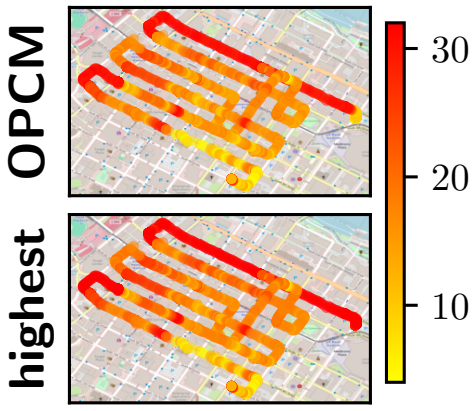


Figure 5.17: Evaluating OPCM compliance with legacy link quality criterion.

loop as Figure 5.3. We pre-configure the cell set $\mathcal{C}$, turn off cell set pruning and delayed reconfiguration, and use a special code (∗#2263#) to switch cell combinations. Figure 5.17 plots live video streaming's bitrate for OPCM (top plot) and compares it with the highest performance achieved by any of the other cell combinations **S1-4** (bottom plot). The results show that OPCM operates close to the highest performing setting: the median bitrate gap between the two is only 2.4 Mbps (10.1%). A small gap is because OPCM's epsilon-greedy policy starts with zero knowledge of cell combinations' performance. In comparison, the legacy CM had a median bitrate gap of 70.1% with the highest performing setting (see Figure 5.3b).

5.8.4 End-to-end System Evaluation

OPCM satisfies all RAN objectives. Here, our setup only utilizes virtual UEs with *sload* between 40–80%. We plot user fairness (defined in §5.4.2) and spectral efficiency (bit/s/Hz), which indicates the amount of information sent through a network using the available bandwidth. We normalize spectral efficiency by dividing it by the highest value of the respective cell. Figure 5.18 yields two key takeaways. First, since iCellSpeed does not respect RAN objectives, it costs

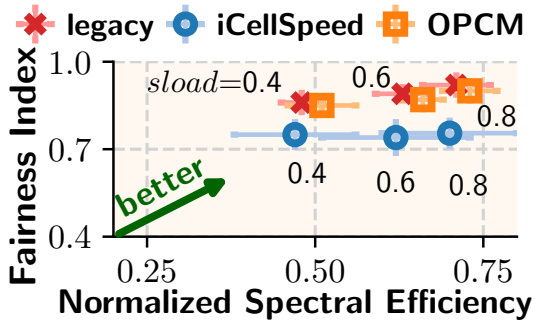


Figure 5.18: Comparing RAN metrics across various load conditions.

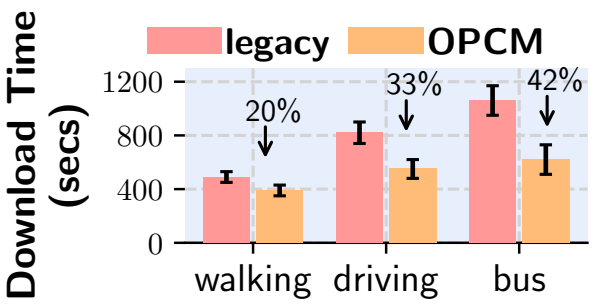


Figure 5.19: OPCM performance under different mobility scenarios.

12.7–17.9% in terms of fairness. Moreover, it improves the spectral efficiency of high-bandwidth cell combinations, while the efficiency of others degrades (see high variation of iCellSpeed's spectral efficiency). In other words, iCellSpeed overloads some cells while others are underused. Second, OPCM's fairness is within 98–99% of legacy. In addition, OPCM improves spectral efficiency by 2–3% compared to the legacy. Although not shown, the load distribution index of OPCM is 0.91–0.94 for different $sload$ values.

OPCM is particularly useful under high mobility. Remember that each virtual UE's channel trace (see §5.8.1) belongs to a specific mobility scenario (e.g., walking, driving, bus). To compare OPCM gains across mobility scenarios, we configure all virtual UEs to repeatedly download a 256 MB file from the remote server for 8 mins. To plot results, we form UE groups based on the mobility scenario of UEs' channel traces. Figure 5.19 shows the average file download time of UE groups. Compared to the legacy case, OPCM reduces the average file download time for *bus* UEs by 41.5%. In contrast, *walking* UEs only see 20.4% reduction. When active cell combination's performance fluctuates rapidly (i.e., *bus*) and the throughput gap between the active and the highest performing cell combination widens, OPCM is more likely to select a better cell combination at

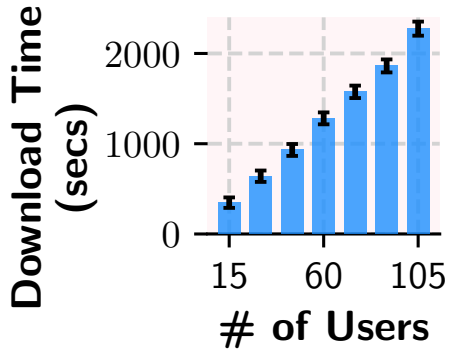


Figure 5.20: OPCM scalability as the number of users increases.

Table 5.4: Comparing system overhead (30 users).

Metric	Legacy	iCellSpeed	**OPCM**
CPU Utilization (%)	35.9 ± 4.8	41.1 ± 5.4	42.4 ± 5.6
Memory Utilization (%)	17.8 ± 3.1	33.7 ± 4.1	20.9 ± 3.6

the next time step due to its greedy policy (see §5.4.3). This leads to higher OPCM gains under complex mobility scenarios.

OPCM is scalable and involves lightweight communication and system costs. First, we evaluate OPCM under large-scale users. Figure 5.20 plots the file download time of a 256 MB file as the total number of users (virtual UEs) grows. As users increase, the download time gradually increases due to the limited bandwidth. However, the increase is almost linear, and the standard deviations are small, suggesting that OPCM offers application-level fairness in the presence of multi-user competition.

Second, we record OPCM's CPU and memory consumption in Table 5.4. It shows that compared to legacy, OPCM increases CPU and memory utilization by 6.5% and 3.1%, respectively. Although not shown, the CPU utilization only increases slightly (2.2%) when users go from 30 to 90. Although not a fundamental limitation, iCellSpeed incurs higher memory usage than OPCM because its *iProfile* module tracks more per-UE state, reflecting both the frequency and performance of cell choices.

Third, OPCM slightly increases the signaling overhead between UE and BS due to exploration. In the average case, the number of signaling messages increases by 11.6% (from 68 to 76 per

minute) compared to the setup where exploration is disabled on our testbed. Similar to legacy CM, this overhead is proportional to UE mobility: faster-moving UEs experience quicker channel variations and more frequent changes in the best cell combination, leading to higher signaling rates.

OPCM efficiently manages advanced CA/DC settings. Using large-scale *ns-3* simulations, we evaluate OPCM with advanced 4G/5G numerologies and variable number of cell combinations. While not shown, our results yield two insights: *(i)* despite five cell combinations with advanced CA and DC settings, OPCM maintains fairness within the δ_b^{FI} range ($FI > 0.85$), evenly distributing load and achieving >90% normalized spectral efficiency across all cells; and *(ii)* increasing the number of cell combinations has diminishing returns, e.g., increasing cell combinations from 3 to 5 only improves average file download time by 9.1%. For comparison, moving from one to three cell combinations significantly enhances download time by 26.0%.

5.8.5 Micro-benchmarks

Decision Framework vs. Oracle. We use *ns-3* simulations to compare OPCM *Decision Framework* with the *Oracle*, which has the complete knowledge of cell combinations' performance and does not require exploration. It thus represents a performance upper-bound. We turn off *Profiling Engine*, disable the delayed reconfiguration mechanism, and use ground-truth performance predictions. Figure 5.21 yields three main insights. (i) OPCM's epsilon-greedy policy effectively balances the exploration and exploitation tradeoff. OPCM is within 98.4% and 83.7% of the *Oracle* in terms of fairness and file download time, respectively. This gap can be attributed to exploration, especially when OPCM starts building performance estimates (see §5.8.3). (ii) The

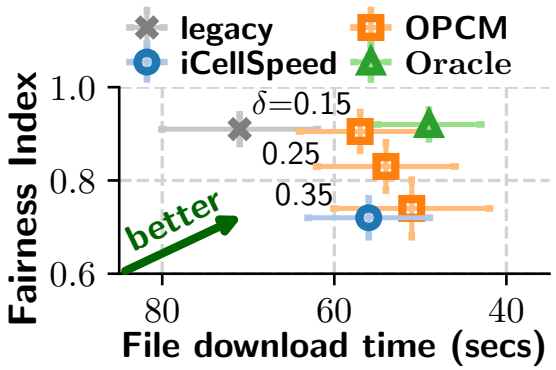


Figure 5.21: Decision framework vs. Oracle.

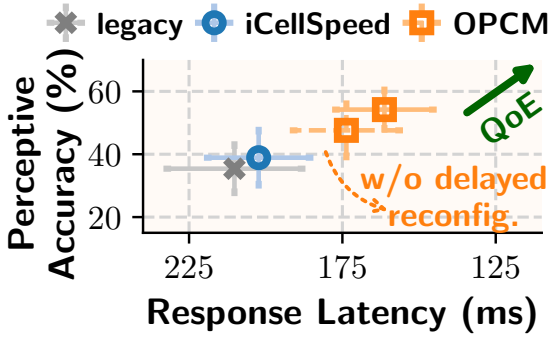


Figure 5.22: Comparing OPCM execution module with legacy.

objective-aware *Decision Framework* offers adjustable knobs (δ) to balance the tradeoff between application performance and RAN metrics. (iii) When RAN policies are more tolerant (i.e., large δ), OPCM behaves similar to iCellSpeed.

Data-plane Interruption Reduction in OPCM. Our results in Figure 5.22 show that OPCM achieves 7.7% lower average response latency and 12.2% higher perceptive accuracy compared to the case when queuing-aware delayed reconfiguration is disabled on OPCM. It drains the uplink/downlink queues before triggering a CM procedure, reducing the queuing delay for video frames. This ultimately leads to a lower response latency and better perceptive accuracy.

OPCM Profiling Engine vs. Baselines. Using the full might of our *corpus*, we compare OPCM with three baselines: (i) *Profiling Engine* without time-lagged cross-correlation (TLCC), (ii) iCellSpeed, and (iii) iCellSpeed with TLCC. The root mean square error (RMSE) and mean absolute error (MAE) between estimated and ground-truth throughput is normalized by each UE's mean throughput. Apart from that, we also compute the ranking accuracy (RA) that measures how accurately an approach predicts the highest performing cell combination. Table 5.5 presents the summary of our results. There are four main takeaways: (i) performance estimation can tolerate

Table 5.5: OPCM Profiling Engine vs. Baselines.

Metric	OPCM	OPCM w/o TLCC	iCellS	iCellS w/ TLCC
RMSE	0.15 ± 0.04	0.33 ± 0.06	0.25 ± 0.06	0.13 ± 0.03
MAE	0.12 ± 0.03	0.24 ± 0.06	0.18 ± 0.03	0.10 ± 0.02
RA	0.94 ± 0.02	0.74 ± 0.06	0.82 ± 0.04	0.94 ± 0.03

small errors since we only need to know the comparative performance (ranking) of cell combinations. Notice that although the RMSE is 0.13–0.33 depending on the approach, RA is high (0.94–0.74); (ii) TLCC across cell combinations improves RA (even for iCellSpeed) by up to 27.0%; (iii) OPCM achieves 14.6% higher RA than iCellSpeed on average. This is because OPCM can utilize passive approximation to build more accurate performance estimates; and (iv) adding TLCC to iCellSpeed makes it achieve slightly (13.3%) lower RMSE than OPCM because of its *iProfile* module that builds more precise profiles but also results in higher memory usage (Table 5.4).

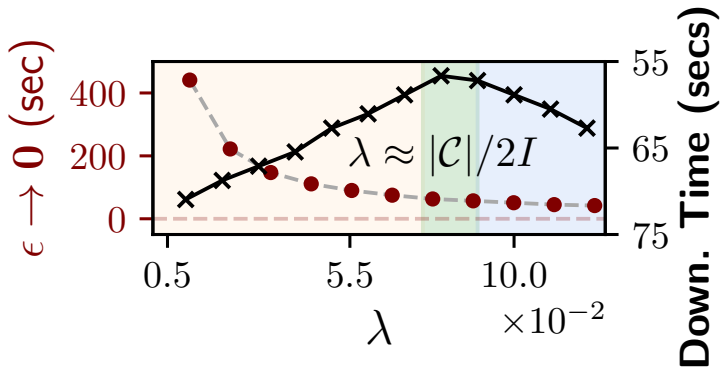


Figure 5.23: Impact of λ on OPCM performance.

Decay rate for ϵ-greedy policy. The decay rate λ controls how fast exploration rate ϵ decreases at startup. We run *ns-3* simulations using our dataset from §5.2.3. Figure 5.23 plots how quickly ϵ decays to 0 for different λ values (left y-axis). It also shows the average file download time (right y-axis inverted). When λ is too small (yellow region), OPCM does not explore enough at startup to find the best performing cell combination. In contrast, when λ is large (blue region),

OPCM has lower network throughput due to frequent exploration. OPCM performs best when λ is close to $|\mathcal{C}|/2I$=0.08 (green region). This value is not surprising as the optimal λ value [85] for an epsilon-greedy policy is: (i) proportional to the number of cell combinations to explore $|\mathcal{C}|$, and (ii) inversely proportional to the best cell combination change frequency, which is I=32 secs on average for our dataset.

5.9 Discussion & Conclusion

Deployment Scope. OPCM's gains are proportional to the diversity of available cells, making it most effective in dense urban deployments. In suburban or rural areas with fewer alternatives, OPCM naturally converges to legacy CM decisions without added cost. The framework adapts to permanent changes: new cells are incorporated as soon as UEs report them, while cell failures are handled using the same fallback and recovery mechanisms already present in standard CM workflows.

Mobility and Coordination. When a UE connects to a legacy BS that does not run OPCM, our backward-compatible design ensures a seamless fallback to legacy CM. Performance can be further improved by sharing profiling information and performance summaries across neighboring BSs over X2/Xn interfaces. Integration with O-RAN Radio Intelligent Controllers (RICs) [127] is a natural extension: RICs could aggregate cross-site profiles and policies to enable richer, network-wide coordination and policy enforcement.

OPCM Time Step. OPCM uses a default 1-second time step Δt to balance responsiveness with signaling and computation overheads. This choice provides timely adaptation for

most mobile scenarios while keeping control-plane activity modest. The time step is operator-configurable: smaller values let the system react faster (at higher overhead), while larger values reduce cost but slow adaptation. Developing adaptive time-stepping policies that tune Δt based on observed volatility or operator preference is a promising direction for future work.

Limitations and Future Work. We note several limitations that suggest fruitful follow-ups. First, OPCM currently focuses on BS-local decision-making; extending to coordinated multi-BS optimization (explicitly modeling cross-link interference) would enable further gains in dense deployments but requires careful cross-site policy design. Second, while our passive, correlation-aware profiling reduces probing overhead, there are corner cases (extreme mobility, abrupt load shifts) where active exploration remains necessary — devising safe exploration schedules that guarantee QoE is an open problem. Third, although we evaluated OPCM across many realistic settings, validating it at operator scale (with commercial core networks and large user populations) would strengthen deployment claims. Finally, tighter integration with O-RAN workflows (RIC apps, xApps/ rApps) and support for emerging NUMS/AI-driven RAN control loops present valuable engineering opportunities.

Concluding Remarks. We expose a new optimization dimension in the 5G/NextG ecosystem: performance-driven connectivity management. Our multi-country measurements demonstrate wide availability and heterogeneity of 4G/5G cell deployments and quantify the missed application-level performance opportunities under legacy CM. OPCM provides a practical, operator-friendly framework that (i) decouples performance objectives from legacy CM procedures, (ii) enforces RAN policies and fairness, and (iii) scales via hybrid profiling (active + passive correlation-aware estimation) with modest overhead. Implemented on pragmatic open-source stacks and

evaluated in over-the-air and trace-driven settings, OPCM shows that intelligent, policy-aware CM services are feasible and valuable for modern cellular networks.

Chapter 6

Energy-Aware Idle Measurement Adaptation

6.1 Introduction

The rapid evolution of mobile networks has enabled unprecedented connectivity, ultra-low latency, and high data rates. At the same time, energy efficiency for user equipment (UE) has become increasingly critical—not only to prolong battery life, but also to support sustainability goals as billions of devices remain persistently connected [79, 57]. While connected-mode transmissions often dominate performance studies, a substantial portion of a UE's lifetime is spent in idle or inactive states. During these states, the UE performs essential radio resource management (RRM) procedures, including intra- and inter-frequency measurements and paging decoding, to support mobility and efficient resumption of data services. Although necessary for reliability, these periodic measurement activities contribute significantly to idle-mode power consumption, making them a promising target for optimization.

To reduce idle-mode energy cost, 3GPP has introduced measurement relaxation mechanisms that suppress certain measurements under low mobility or non-cell-edge conditions. While effective in specific scenarios, these mechanisms remain conservative and limited in scope. Through

our experiments, we observe many practical situations in which channel conditions evolve in a predictable manner over time. In such cases, measurement prediction can be used to further relax RRM activity and reduce energy consumption beyond what current standards permit. However, existing UEs cannot exploit this predictability, leading to missed opportunities for energy savings. For example, inter-frequency measurements are frequently performed despite long intervals of stable serving-cell quality, causing repeated modem wakeups (Section 6.1.1).

Existing research has largely focused on optimizing connected-mode mobility procedures or broader network-level energy savings [10, 110, 144, 33], leaving idle-mode measurement control comparatively underexplored. This gap is particularly timely given the capabilities of modern devices. Contemporary smartphones possess significantly greater computational resources and richer sensor context (e.g., inertial sensors, GNSS), enabling more sophisticated prediction mechanisms beyond the limited mobility-based heuristics defined in 3GPP. Importantly, a UE-side solution requires no network involvement and allows vendor-specific innovation without compromising standard compliance.

In this chapter, we propose an adaptive measurement relaxation framework PARMA that leverages channel prediction and context-awareness to reduce unnecessary idle-mode measurements. The key insight is that measurement activity exhibits temporal structure shaped by mobility and radio conditions. By predicting channel evolution and quantifying the risk of missed cell reselection events, our approach selectively suppresses inter-frequency measurements while preserving reliability guarantees.

To support practical deployment, we introduce a lightweight online energy estimation method based solely on measurement configuration profiling, avoiding the need for continuous external power instrumentation. Furthermore, we design a simulator that faithfully mimics UE idle-mode

measurement behavior using real UE traces, enabling controlled evaluation of adaptive relaxation strategies. We empirically characterize idle-mode energy costs and evaluate relaxation policies using real-world traces collected via Monsoon Power Monitor [114] and Samsung's Shannon Diagnostic Monitor (SDM) for Exynos chipsets [146].

Our evaluation demonstrates that adaptive relaxation can unlock substantial energy savings. Specifically, we observe up to 16.6% average reduction in idle-mode power consumption compared to baseline behavior, with upper-bound gains approaching 20% when inter-frequency measurements are aggressively suppressed. In our tested scenarios, a lightweight decomposition-based prediction model consistently outperforms one-dimensional convolutional neural network (CNN1D) and long short-term memory (LSTM) baselines, achieving lower prediction error even under higher relaxation levels. While we leave broader model exploration for future work, these results indicate that simple, interpretable models are sufficient in this setting.

In summary, this chapter makes three contributions: ***(i)*** an empirical characterization of idle-mode measurement energy costs, ***(ii)*** a prediction-driven relaxation scheme with a formalized cost–utility tradeoff between energy savings and mobility reliability, and ***(iii)*** a simulator-driven evaluation framework validated with real-world measurement traces.

6.1.1 Motivating Prediction-Aware Measurement Scheduling

Data Collection. We collected high-resolution power traces using the Monsoon High Voltage Power Monitor [114], which captures fine-grained voltage and current samples sufficient to isolate the energy cost of individual paging occasions and measurement events. To correlate power

consumption with lower-layer operations, we extracted cellular logs using Samsung's Shannon Diagnostic Monitor (SDM) on a Galaxy S24+ equipped with the Exynos 2400 chipset [145].

The combination of power traces and protocol-layer logs allows us to attribute energy consumption directly to paging decoding, intra-frequency measurements, and inter-frequency scans. Additional implementation and experimental details are provided in Section 6.3.1.

6.1.2 Limitations of Existing Measurement Relaxation

Although 3GPP-defined measurement relaxation reduces idle-mode energy under certain conditions, our measurements reveal substantial untapped opportunities. These inefficiencies are not due to misconfiguration but stem from inherent limitations of static, threshold-based mechanisms that cannot capture fine-grained temporal dynamics in real-world radio environments.

Current relaxation frameworks rely on coarse indicators such as mobility classification derived from cell reselection frequency and static RSRP variation thresholds (e.g., classifying a UE as low mobility if RSRP variation remains within 3 dB for one minute, as observed in our experiments) [7]. This network-centric approach overlooks the rich temporal context available at the UE, including high-resolution signal histories and environmental cues. As a result, UEs may continue performing full measurement cycles even when channel evolution would permit safe relaxation.

We next present two representative real-world scenarios demonstrating missed energy-saving opportunities.

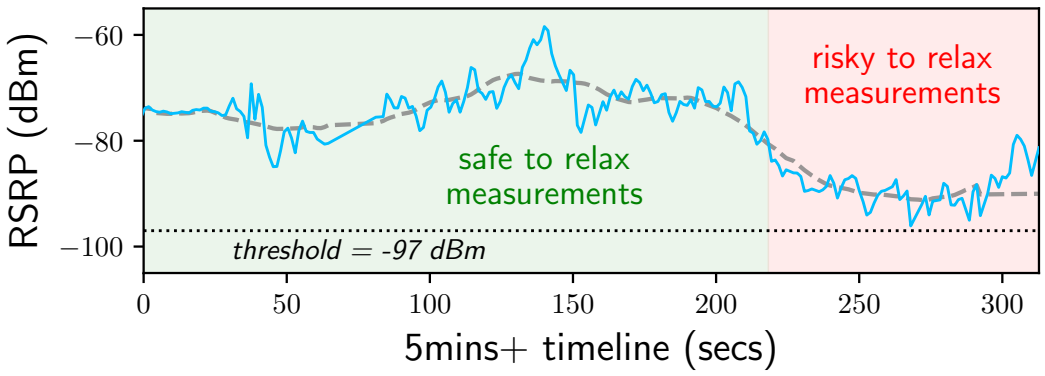


Figure 6.1: RSRP trace from a walking experiment. Green: strong, stable signal; red: degrading region. Dashed gray line shows underlying trend; dotted line marks reselection threshold (-97 dBm).

6.1.2.1 Predictable Mobility with Short-Term Fluctuations

To evaluate additional relaxation opportunities beyond 3GPP rules, we conducted a controlled walking experiment around a building complex while holding the device in hand. Figure 6.1 shows the measured RSRP over more than five minutes. The network-configured reselection threshold was -97 dBm (dotted line).

During the first two-thirds of the experiment (green-shaded region), the RSRP remained 12.1–38.6 dB above the reselection threshold. Under such conditions, relaxing inter-frequency measurements would not compromise reselection responsiveness. However, the 3GPP low-mobility condition was never triggered because short-term fluctuations exceeded the 3 dB bound—even though the overall signal evolution followed a clear and gradual trend (dashed gray line).

This example highlights a fundamental limitation: static fluctuation thresholds treat short-term variance as mobility, even when long-term signal evolution is predictable and stable. A trend-aware relaxation mechanism could safely suppress measurements during such predictable intervals.

In the later portion of the trace (red-shaded region), signal quality degraded toward the reselection threshold. In this regime, aggressive relaxation becomes risky. However, the underlying trend provides early warning of degradation. A predictive system could resume full measurement frequency proactively before entering high-risk zones, preserving reliability while still benefiting from earlier relaxation.

This example demonstrates that threshold-based rules are overly conservative in predictable mobility scenarios with short-term variability. A prediction-aware mechanism can aggressively relax measurements during stable, high-margin periods, and anticipate degradation and resume measurements before reselection risk rises.

Similar patterns occur in slow vehicular motion under strong coverage or stationary users experiencing environmental fluctuations. In all such cases, predictive relaxation offers energy savings beyond rule-based heuristics.

6.1.2.2 Stable Serving Cell with Higher-Priority Neighbors

We next examine a stationary experiment inside an office building. Figure 6.2 shows RSRP traces for multiple visible cells. The serving cell (solid light-gray line) operates on Band N25, while higher-priority N41 neighbors (dashed lines) remain consistently below the reselection threshold of -100 dBm.

According to 3GPP specifications [10], the UE must periodically measure higher-priority inter-frequency carriers unless relaxation conditions are met. In this scenario:

- The serving cell remains strong and stable.
- All higher-priority neighbors remain well below threshold.

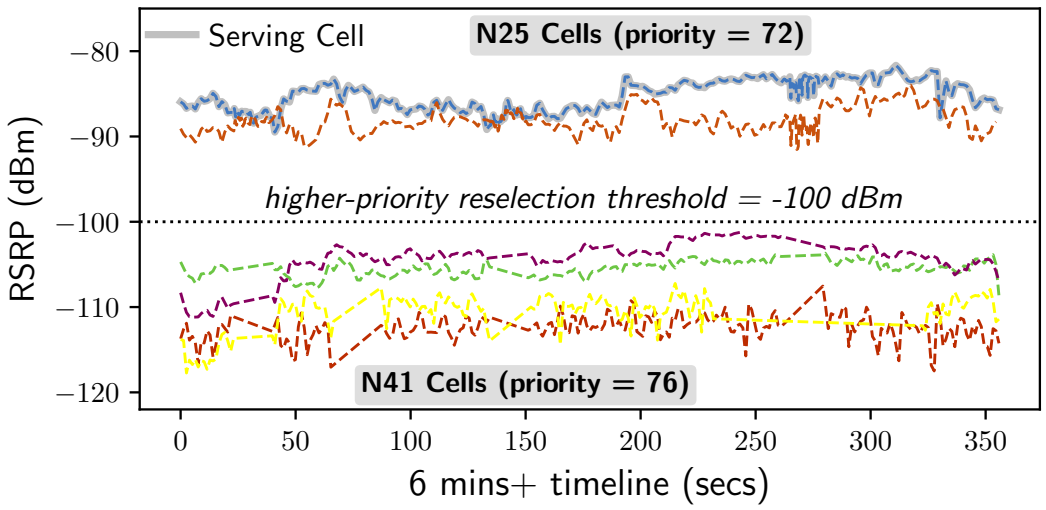


Figure 6.2: Stationary experiment. Serving cell (N25) remains strong; higher-priority N41 neighbors remain below reselection threshold. Continuous inter-frequency measurements occur without reselection events.

- No reselection events occur.

Despite this, the UE continues performing inter-frequency measurements for extended periods (over one hour in our tests). These scans incur repeated RF retuning and measurement gaps, significantly increasing idle-mode energy consumption without any mobility benefit.

This scenario reveals another structural limitation of the current framework: persistent measurement of consistently weak higher-priority neighbors. A context-aware system could detect prolonged stability and suppress redundant inter-frequency scans, reactivating them only when signal trends indicate potential change.

6.1.3 Towards Adaptive Measurement Relaxation

The above scenarios show that the problem is not misconfiguration of static thresholds; rather, static rules cannot adapt to diverse and evolving contexts. Even optimal parameters for one scenario will be suboptimal in another.

An adaptive relaxation mechanism must address two key questions:

- **Predictability:** Can the UE forecast near-future signal quality with sufficient confidence based on recent history and context?
- **Risk–Utility Tradeoff:** Given predicted signal evolution, is the probability of missing a reselection event sufficiently low to justify skipping measurements?

By integrating signal prediction and utility estimation into relaxation decisions, the UE can selectively suppress measurements based on actual risk rather than static heuristics. This complements existing 3GPP mechanisms rather than replacing them.

Recent 3GPP Release 19 discussions introduce AI/ML-assisted enhancements for RRM, including measurement prediction and event forecasting [138]. These developments underscore the timeliness of prediction-based relaxation mechanisms. Our work aligns with this direction while remaining fully UE-side and backward-compatible.

6.2 Proposed Solution

We now present **P**rediction-based **A**daptive **R**RM **M**easurement **A**daptation (PARMA), designed to reduce idle-mode energy consumption without sacrificing cell reselection performance. At a high level, the solution skips $R - 1$ consecutive measurement cycles for a relaxation factor R, predicting the skipped measurements instead of performing them physically (see Figure 2.11). While 3GPP defines certain relaxation mechanisms for low-mobility or high-signal-quality scenarios, these are non-adaptive, lacking the intelligence to proactively skip measurements when channel conditions are predictable. Furthermore, the trade-off between energy savings and cell

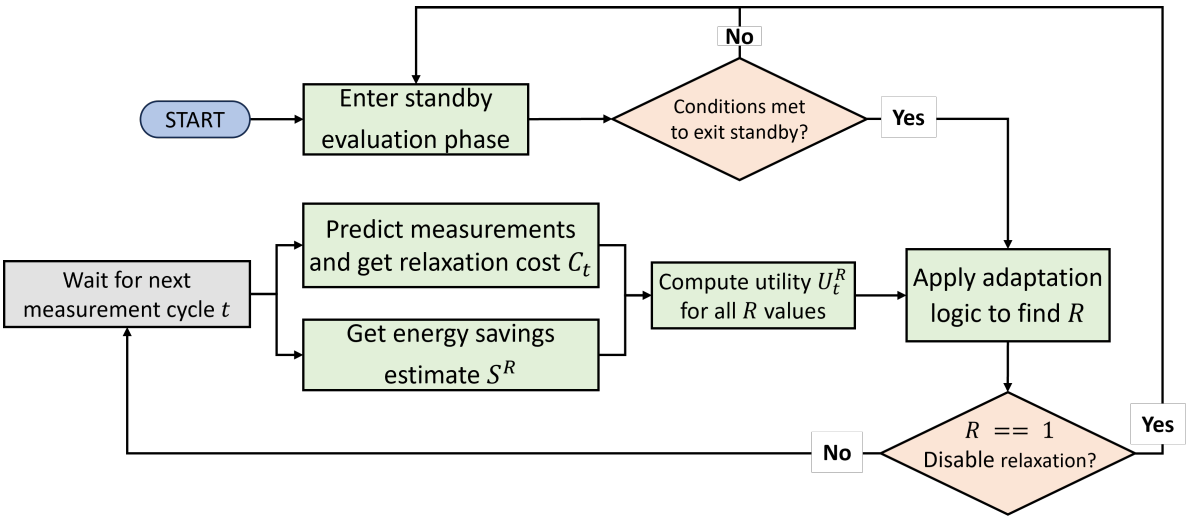


Figure 6.3: High-level workflow of the adaptive measurement relaxation framework, showing its five core components and their interactions.

reselection performance is delicate: aggressive skipping can risk missing cell reselection threshold crossings, while an overly conservative behavior forfeits potential energy savings.

The design must therefore address three key challenges:

- Predict future signal quality measurement values with sufficient accuracy to safely skip physical measurements.
- Estimate the energy savings achievable under the current measurement configuration.
- Balance energy savings against prediction errors and reselection risks while dynamically adapting the relaxation aggressiveness as radio conditions change.

To tackle these challenges, we design five core components as shown in Figure 6.3. First, the *measurement prediction* module (Section 6.2.1) forecasts future signal quality from historical observations, handling varying relaxation factors and missing data due to skipped measurements. It outputs a relaxation cost that combines prediction error, reselection risk, and computation

overhead for use in the utility computation stage. Second, the *energy estimation* module (Section 6.2.2) quantifies the expected power savings from skipping measurements under the current configuration via lightweight configuration-based modeling. Third, the *utility computation* module (Section 6.2.3) combines projected savings with prediction costs—including both performance risk and computation overhead—to evaluate the net benefit of relaxation. Fourth, the *adaptive relaxation mechanism* (Section 6.2.4) dynamically tunes R over time to maximize the utility while reacting to changing channel conditions. Finally, the *standby evaluation* module (Section 6.2.5) handles cold-start scenarios with limited historical data as well as other conditions where measurement relaxation cannot be reliably applied.

6.2.1 Measurement Prediction and Relaxation Cost

The first component of our framework is the *measurement prediction* module, which forecasts future signal quality values while quantifying the prediction error and the associated cell reselection risk. The central principle is straightforward: measurements can be safely relaxed when predictions are accurate and the predicted signal quality remains comfortably distant from reselection thresholds, whereas high prediction error or proximity to thresholds warrants more frequent measurements. In addition to these two factors, the module also accounts for the computational and memory overhead of running the prediction pipeline itself. The final output of this module is the *relaxation cost*, which combines prediction error, reselection risk, and solution overhead, and serves as an input to the utility computation stage.

Let R denote the current measurement relaxation factor, where the UE performs a physical measurement once every R paging cycle and predicts the measurement values in the intervening

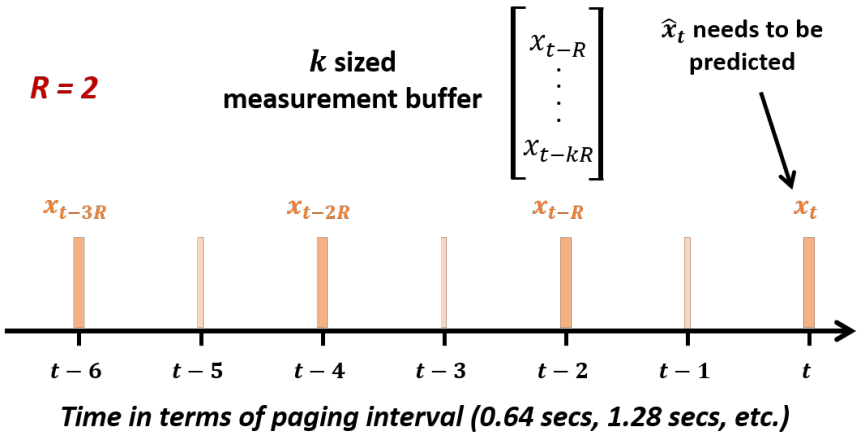


Figure 6.4: Collection of historical measurements aligned to the relaxation factor to form the measurement buffer used for prediction.

cycles (see Figure 6.4 for a detailed example). Let t denote the current paging cycle when a measurement is scheduled; the most recent physical measurement then occurred at $t - R$. The prediction module maintains a measurement buffer:

$$X_{(t-R)} = \{x_{(t-kR)}, \ldots, x_{(t-R)}\}, \tag{6.1}$$

where $x_{(t-iR)}$ represents the measured signal quality (e.g., RSRP, RSRQ, or both) i relaxation intervals ago. The signal quality values $x_{(t-iR)}$ are normalized* between 0 and 1 to account for differences in units and dynamic range, ensuring that error values are comparable across signal quality metrics and scenarios. The buffer $X_{(t-R)}$ is then fed into a prediction model $f(\cdot)$, which outputs the predicted signal quality for the upcoming cycle:

$$\hat{x}_t = f\left(X_{(t-R)}\right) \tag{6.2}$$

*For instance, we apply min–max scaling using -156 dBm as the minimum and -31 dBm as the maximum reference levels to normalize RSRP values.

Since R is dynamically adapted by the *adaptive relaxation mechanism* (Section 6.2.4), the historical measurement data used for prediction may not be aligned to the current relaxation factor. This introduces two distinct data preparation scenarios:

- **Increase in R:** When the relaxation factor increases, the spacing between past measurements in the buffer becomes smaller than the current R. In this case, the buffer can be realigned by selectively dropping older measurements to match the expected temporal spacing.

- **Decrease in R:** When the relaxation factor decreases, the available history may be too sparse to match the new, shorter interval. In this case, intermediate samples must be reconstructed using techniques such as linear interpolation, forward-filling (replicating the last known value), or extrapolating trends from recent history. In low-confidence cases, conservative estimates may be used to reduce prediction risk. We use forward-filling in our design.

These choices represent reasonable design options for handling misaligned measurement histories, though other alignment strategies could also be applied. A systematic comparison of such alternatives is left to future work.

In environments where the channel exhibits both long-term trends and short-term fluctuations, directly predicting x_t from raw historical values can be suboptimal. To improve robustness, the module can decompose the signal into *trend* and *residual* components: $x_t = \tau_t + \rho_t$, where τ_t

is the smoothed long-term trend (obtained via moving average) and ρ_t is the short-term residual. The prediction process then proceeds independently for each component:

$$\hat{\tau}_t = f_\tau(\tau_{(t-kR)}, \ldots, \tau_{(t-R)}); \tag{6.3}$$

$$\hat{\rho}_t = f_\rho(\rho_{(t-kR)}, \ldots, \rho_{(t-R)}); \tag{6.4}$$

$$\hat{x}_t = \hat{\tau}_t + \hat{\rho}_t. \tag{6.5}$$

This approach reduces the likelihood of *adaptive relaxation mechanism* from overreacting to transient noise while still preserving responsiveness to sustained signal quality changes. In particular, it enables accurate anticipation of threshold crossings in sparse-measurement situations, thereby improving the reliability of relaxation decisions. To realize this capability, we design a lightweight prediction module whose architecture is described next.

Model Architecture. The architecture of our prediction module, illustrated in Figure 6.5, begins with the measurement buffer that stores the most recent k historical measurements, spaced R paging cycles apart. This buffer forms the unified input to our decomposition-based prediction pipeline. The trend is obtained via the moving average smoothing operation to capture gradual variations over extended periods, while the residual isolates higher-frequency fluctuations that may still be relevant for reselection decisions.

As shown in Figure 6.5, the decomposed components are then fed into two independent linear predictors: f_τ for the trend branch and f_ρ for the residual branch. The trend predictor leverages its smoothed input to extrapolate slow signal drifts with high stability, whereas the residual predictor models rapid but short-lived deviations. The outputs $\hat{\tau}_t$ and $\hat{\rho}_t$ are then linearly combined to yield the final prediction $\hat{x}_t$. This modular design improves robustness in sparse-measurement

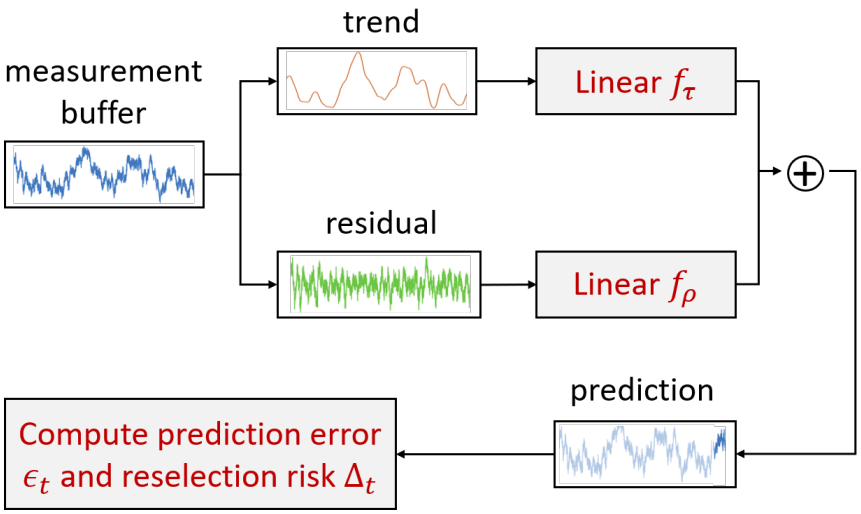


Figure 6.5: Decomposition-based prediction module. Measurements from the buffer are split into trend and residual components, predicted separately, and then combined to produce $\hat{x}_t$.

conditions (large R values) by preventing the *adaptive relaxation mechanism* from overreacting to transient noise, while still enabling timely detection of threshold crossings.

Prediction Error. Once the ground-truth measurement x_t becomes available (i.e., when a physical measurement is taken), the prediction error is computed as: $\epsilon_t = |x_t - \hat{x}_t|$, where $\epsilon_t \in [0, 1]$ represents the normalized prediction error. A high ϵ_t indicates reduced prediction reliability and may warrant lowering the relaxation factor R to increase measurement frequency, and vice versa. To improve stability, the module maintains an exponentially weighted moving average (EWMA) of prediction errors:

$$\epsilon_t = \alpha \cdot \epsilon_{(t-R)} + (1 - \alpha) \cdot |x_t - \hat{x}_t|, \tag{6.6}$$

where $\alpha \in [0, 1]$ controls the smoothing factor and filters out transient spikes.

Reselection Risk. We approximate the risk of missing reselection opportunity by a function of the gap to the reselection threshold. A sigmoid form is chosen since it smoothly interpolates

between low and high risk, while emphasizing rapid growth in risk as the predicted value approaches the threshold. Particularly, we define the risk as:

$$\Delta_t = \left(1 + e^{(\hat{\tau}_t - T)/\gamma}\right)^{-1},$$

where $T \in [0, 1]$ is the network-configured measurement or reselection trigger threshold. Smaller Δ_t values correspond to lower risk (signal well above the threshold), while larger values indicate higher risk of crossing the threshold, increasing the likelihood of missed reselections if measurements are skipped. Additionally, we use trend $\hat{\tau}_t$ instead of the actual signal $\hat{x}_t$ as the noise can render Δ_t unstable. Note, however, that the residual component is still combined with the trend when computing the prediction error ϵ_t. Similar to prediction error, a smoothed estimate is maintained:

$$\Delta_t = \beta \cdot \Delta_{(t-R)} + (1 - \beta) \cdot \left(1 + e^{(\hat{\tau}_t - T)/\gamma}\right)^{-1} \tag{6.7}$$

Solution Overhead. This term captures the computational, memory, and energy cost of running the prediction pipeline itself. While generally lightweight, this cost ϕ can become significant in scenarios with frequent predictions or constrained UE resources.

Relaxation Cost. The final output of the measurement prediction module is the *relaxation cost* $C_t \in [0, \infty]$, which combines the three factors above. We adopt a linear formulation, as it keeps the design lightweight, interpretable, and easy to tune: prediction error ϵ_t and reselection risk Δ_t contribute as distinct terms with clear roles, while the formulation remains flexible enough to incorporate additional factors if needed. Formally,

$$C_t = \lambda \epsilon_t + \eta \Delta_t + \phi, \tag{6.8}$$

where λ and η are scaling factors that control the relative weight assigned to prediction error and reselection risk, respectively. This cost is then passed to the utility computation stage (Section 6.2.3) to determine whether the current relaxation factor R should be maintained, increased, or decreased. Although the notation does not make it explicit, C_t also depends on the relaxation level: with larger R, the prediction window $\delta t = t - (t - R)$ becomes longer, making accurate prediction more challenging and thereby increasing the effective cost.

The parameter λ directly influences how aggressively the system reacts to *prediction inaccuracy*. A large λ value amplifies the penalty for high ϵ_t, making the *adaptive relaxation mechanism* more conservative by lowering R quicker when prediction reliability degrades. Conversely, a small λ value reduces the impact of ϵ_t, allowing higher relaxation levels to be maintained despite occasional mispredictions. Similarly, η determines the emphasis placed on *reselection risk*. A large η heavily penalizes situations where the predicted signal quality is close to the reselection threshold, ensuring that the UE reverts to frequent measurements to avoid missed reselection triggers. A smaller η tolerates operation closer to the threshold, enabling more aggressive measurement skipping at the cost of potentially slower reselection responsiveness. Tuning λ and η allows us to trade off *energy efficiency* against *reselection performance*. For instance, $(\lambda, \eta) = (0.1, 0.1)$ biases the system towards maximizing idle-mode energy savings with a higher R, whereas $(\lambda, \eta) = (1.0, 1.0)$ yields a highly conservative strategy that minimizes reselection performance loss but sacrifices some energy gains. The choice of these weights thus reflects the desired operational profile, device constraints, and tolerance for transient signal degradation. The empirically chosen (α, β, γ, λ, η, ϕ) values are described in Table 6.1.

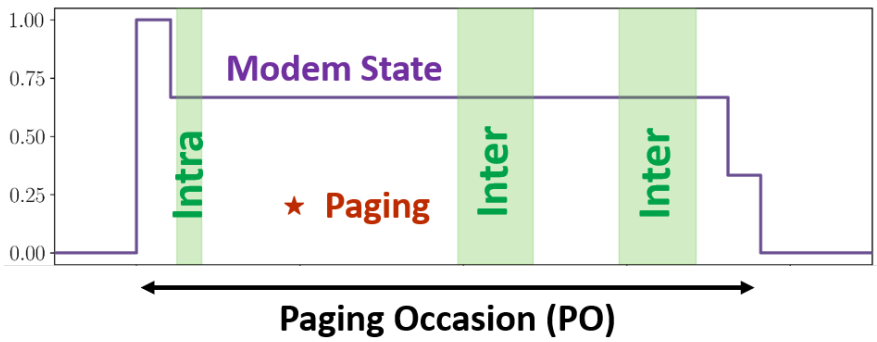


Figure 6.6: Intra- and inter-frequency measurements within a PO.

6.2.2 Energy Saving Estimation

The second core component of the adaptive measurement relaxation framework is the *energy savings estimation module*, which evaluates the potential reduction in idle-mode power consumption when selectively skipping measurements. Even when channel conditions suggest that prediction is feasible, relaxation should only be applied if it produces meaningful energy savings. This is critical because the benefit of skipping measurements varies significantly depending on the measurement configuration, the type of measurements performed, and their placement relative to the paging occasion.

Figure 6.6 illustrates an example configuration showing *intra-frequency* and *inter-frequency* measurements we observed in our experiments. Measurements associated with serving and intra-frequency cells are typically short, occur on the same carrier, and are often integrated directly into the paging occasion before decoding the paging message. These are essential for maintaining synchronization, time tracking, and frequency drift correction, and therefore offer limited scope for relaxation. In contrast, inter-frequency measurements often span longer durations and require *inter-frequency measurement wait time*, making them far more energy-intensive and better candidates for relaxation. For brevity, we refer to this as *measurement wait time*, which may arise

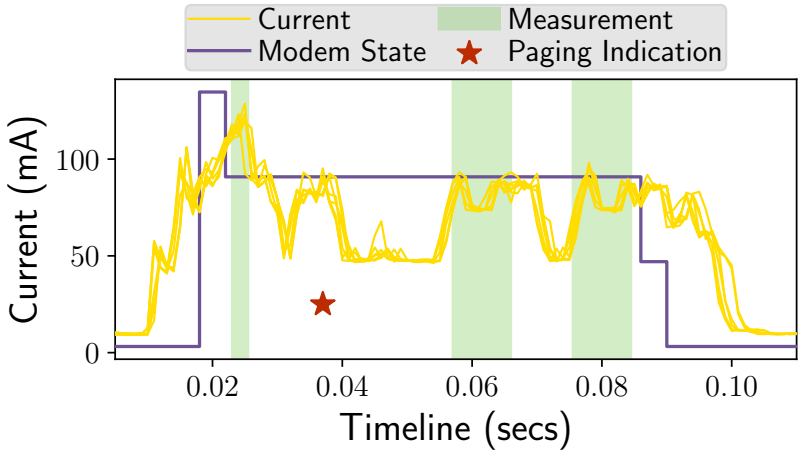


Figure 6.7: Example power traces from Monsoon profiling showing current spikes for measurement events overlayed on top of the PO.

for various reasons, such as retuning the receiver to a different carrier frequency or waiting for the target cell's reference signals.

6.2.2.1 Empirical Power Profiling

One approach to estimating potential energy savings is through *empirical power profiling*. Using tools such as the Monsoon High Voltage Power Monitor [114], we record current traces at 0.2 ms granularity under various measurement configurations across multiple paging cycles. By aligning the recorded current spikes with specific measurement events, it becomes possible to quantify the precise energy cost of each measurement type and *wait time*. Figure 6.7 illustrates such profiling, where distinct current peaks correspond to specific measurement operations. The resulting power model can then be applied at runtime to match the current configuration against pre-profiled scenarios, yielding highly accurate savings estimates.

While this method provides fine-grained accuracy, it has practical limitations. Empirical profiling requires access to specialized measurement equipment, controlled test conditions, and repeated profiling for different devices, bands, and configurations. Furthermore, it is not always

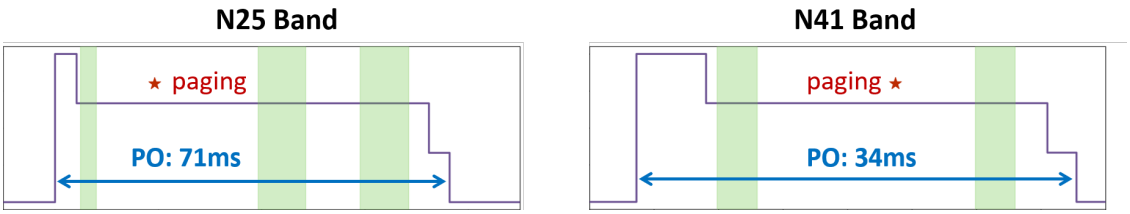


Figure 6.8: PO durations and per-PO energy savings for bands N25 and N41.

feasible to maintain an up-to-date empirical model across all possible network configurations that a UE may encounter in the field. These constraints motivate a more lightweight and generalizable alternative.

6.2.2.2 Configuration-Based Estimation

To overcome the above-mentioned limitations of the empirical power profiling approach, we employ a *configuration-based estimation* method that derives energy savings directly from known measurement timing parameters without requiring external profiling. Since measurement durations, *wait times*, and their placement within the paging occasion remain relatively stable over time, we can approximate the per-measurement energy cost using only the RRM configuration.

For example, if the configuration specifies two inter-frequency measurements per paging occasion, each lasting 6 ms on average with an additional 10 ms *measurement wait time*, the expected energy saving can be estimated by summing the measurement duration and *wait times*, and then normalizing the sum by the total PO duration. This approach is lightweight, requires no special hardware, and adapts easily to varying configurations.

Figure 6.8 shows how PO durations, and hence potential savings, vary across bands due to their respective configurations. In band N25, two inter-frequency measurements with large *measurement wait times* result in a PO duration of 71 ms, whereas in band N41, one shorter inter-frequency measurement yields a PO duration of 34 ms. Monsoon measurements confirm the correlation: skipping inter-frequency measurements saves 3.18–3.81 mAs per PO in N25 and 0.64–1.23 mAs in N41. These ranges correspond to best- and worst-case estimates, depending on whether the savings are measured from the end of the paging cycle or from the start of the first inter-frequency measurement within a PO.

At runtime, the lightweight estimation module logs the timing of each measurement operation by recording the interval between the measurement request and its completion. Using this timing, together with known *measurement wait times*, the module computes the per-PO savings $S \in [0, 1)$ normalized by the total PO duration. The expected savings for a given relaxation level R is then:

$$S^R = S \cdot \frac{R-1}{R} = S \cdot r, \quad r \in [0, 1) \tag{6.9}$$

This formulation allows the *adaptive relaxation mechanism* to quantitatively evaluate whether applying measurement relaxation at the current R is worthwhile given the prevailing configuration and channel conditions. Note that S is a function of the configuration, which we expect to be stable for an extended duration.

6.2.3 Utility Computation

The adaptive measurement relaxation mechanism selects the relaxation factor R that maximizes net benefit by computing the *utility* $U_t^R = S^R - C_t$, defined as the difference between projected

energy savings S^R and the cost of applying relaxation C_t. This formulation explicitly quantifies the trade-off between two competing goals: *(i)* minimizing idle-mode energy consumption, and *(ii)* preserving measurement accuracy and reselection performance. Based on $U_t^R \in [-1, 1]$, the *adaptive measurement relaxation mechanism* determines whether the current relaxation factor R should be maintained, increased, or decreased. A positive utility indicates that the expected savings outweigh the relaxation costs, prompting the mechanism to maintain or increase R up to a predefined upper bound $R_{\max}$. This bound serves as a safety constraint, preventing R from growing to impractically large values that could compromise measurement responsiveness. Conversely, if $U_t^R \leq 0$, R may be reduced, potentially down to $R_{\min} = 1$ (no measurement relaxation).

This formulation adapts naturally to changing conditions. When prediction accuracy is high (ϵ_t low) and the link quality is well above the reselection thresholds (Δ_t low), C_t is small, resulting in higher U_t^R and a tendency toward larger R values for greater energy savings. Conversely, in unstable channels or when operating near the cell reselection threshold, ϵ_t and Δ_t grow, increasing C_t and reducing U_t^R, prompting the mechanism to lower R and restore more frequent measurements. By dynamically balancing these terms, the mechanism ensures that relaxation aggressiveness is automatically tuned to prevailing network and channel conditions.

6.2.4 Adaptive Relaxation Mechanism

The next key component of the framework is the *adaptive measurement relaxation mechanism*, which adjusts the relaxation factor R over time based on the computed utility U_t^R. Rather than

solving a global optimization problem over all future time steps, we apply a greedy approach: at each paging cycle, we select the R value that maximizes the instantaneous utility, i.e.,

$$\arg\max_R U_t^R = \arg\max_R \left(S^R - \lambda\epsilon_t - \eta\Delta_t - \phi\right),$$

while avoiding unnecessary prediction costs or measurement accuracy degradation.

6.2.4.1 Progressive Adaptation Strategy

A lightweight approach is to adopt a *progressive adaptation strategy*. The mechanism begins with a conservative relaxation level, such as $R = 2$, which offers immediate energy savings with minimal prediction error. At each update interval, it evaluates the utility U_t^R for only three candidate values: $R - 1$, R, and $R + 1$. If the utility increases, R is incremented to allow more aggressive measurement skipping; if it decreases or becomes negative, R is reduced to restore measurement reliability. This gradual adjustment allows the mechanism to track changing channel conditions while keeping the running cost ϕ low, since only three utilities are computed per update. If the temporal variation can be large, a gating mechanism can be introduced for extra safety by quickly reverting back to $R = 1$, e.g., when the prediction error suddenly jumps.

Limitations. While progressive adaptation is computationally efficient, it may adapt slowly in scenarios where the optimal R is far from the current value, especially if conditions vary. The mechanism may also converge to a locally optimal R rather than the global optimum if intermediate R values yield lower utility, potentially missing higher-utility configurations.

6.2.4.2 Full Search Strategy

To address these limitations, a *full search strategy* evaluates U_t^R for all feasible $R \in [R_{\min}, R_{\max}]$ and selects the one with the highest utility. If the best R is $R_{\min} = 1$ (no relaxation), the mechanism disables adaptive relaxation and reactivates the standby evaluation module. This ensures that relaxation is only applied when it offers a clear net benefit. This ensures that we select the best relaxation factor R under the current conditions.

The full search strategy achieves the best R selection but at a higher computational cost ϕ, which may offset some of the energy savings S^R—especially for large $R_{\max}$ values. In contrast, the progressive strategy has negligible overhead but may adapt more slowly or miss the optimal R. In our experiment, we observe that a moderate $R_{\max}$ value (e.g., $R_{\max} = 10$) can already approach the energy saving gains of unbounded R_{max} values (check out Section 6.3.2 for details). Nonetheless, the progressive approach remains attractive in low-power devices or when rapid computation is critical. For our implementation, we adopt the full search approach with a bounded $R_{\max} = 10$, which keeps the computational overhead low while choosing the best R for current conditions.

6.2.5 Standby Evaluation Module

The adaptive measurement relaxation mechanism enters the *standby evaluation* mode when current conditions indicate that applying measurement relaxation would not provide a net benefit or cannot be executed with sufficient reliability. This can occur under several scenarios, including:

(i) Insufficient history: The measurement buffer does not contain enough samples to support reliable predictions.

(ii) Configuration change: A major change in network measurement configuration (e.g., neighbor list, periodicity, thresholds, etc.) invalidates the current prediction context.

(iii) No relaxation advised: The mechanism determines that the highest-utility relaxation factor is $R = 1$, meaning no skipping should be applied. This could happen due to significant short-term fluctuations or unstable signal trends resulting in high prediction error ϵ_t or elevated reselection risk Δ_t.

When triggered by insufficient history, the module waits until the measurement buffer reaches the required length k to support stable predictions. Once this condition is met, the mechanism: ***(i)*** predicts the next measurement value from historical data, ***(ii)*** computes relaxation cost C_t, projected savings S^R, and resulting utility U_t^R, and ***(iii)*** evaluates whether the best R exceeds 1. If $R > 1$, the mechanism exits the standby state and resumes adaptive operation as shown in Figure 6.3. If the best R remains 1, the module enters a *wait interval* before repeating the standby evaluation to avoid unnecessary computation cost when gains are not available. This interval can be fixed or adaptive—e.g., scaled according to the most recent utility value $|U_t^{R_{\max}}|$—to avoid unnecessary computation when relaxation is clearly non-beneficial, while still enabling rapid reactivation when conditions improve.

6.3 Evaluation

We now evaluate our proposed adaptive measurement relaxation framework using real-world traces and a custom simulator. Our goals are to characterize idle-mode energy costs, quantify the bounds of potential savings, and assess the effectiveness of prediction-driven adaptive relaxation.

Table 6.1: Default parameters used for evaluating adaptive measurement relaxation.

Parameter	Notation	Default Value
History Buffer Length	k	8
Prediction Error Weight	λ	1.0
Distance To Threshold Weight	η	0.3
Smoothing Factors	$\alpha/\beta/\gamma$	0.9/0.5/0.1
Highest Relaxation Level	R_{max}	10

6.3.1 Experiment Setup

Tools and Devices. Power consumption traces were collected using the Monsoon Power Monitor [114], which provides high-resolution voltage and current measurements to capture the energy cost of individual paging occasions and measurement events. Lower-layer cellular activity was logged using the SDM tool on Samsung Galaxy S24+, which is equipped with Exynos chipset [145]. These SDM logs enabled the extraction of serving and neighbor cell measurements, measurement configurations (frequency, priorities, thresholds, offsets), modem power states, and paging activity. Since simultaneous Monsoon and SDM logging was found to introduce significant measurement noise, we performed separate data collection passes: first, multiple SDM logs to profile measurement configurations at a given location, followed by Monsoon experiments to characterize the corresponding power consumption. Each experiment was repeated multiple times to ensure consistency and statistical confidence.

Data Collection Methodology. All experiments were conducted on T-Mobile's standalone (SA) 5G network, as it was the only provider offering SA coverage in our area. Devices were band-locked to SA 5G, with observed cells spanning bands N25 (1930 MHz), N41 (2496 MHz), and N71 (617 MHz). Traces were collected in both stationary and low-mobility scenarios. We use 70% of the collected data for training our prediction model, with the remaining portion reserved for evaluation.

Simulator. To enable controlled and reproducible evaluations, we implemented an idle-mode simulator that emulates UE state transitions, intra- and inter-frequency measurement durations, paging occasions, and *measurement wait times*. The simulator replays collected traces under different configurations. This design ensures that all methods are evaluated over identical channel conditions and measurement configurations. Unless stated otherwise, simulation parameters and prediction model hyperparameters follow the configuration summarized in Table 6.1. We note that the relaxation cost component ϕ and the network-configured threshold T are not design choices: ϕ reflects implementation-specific factors such as hardware and code overhead, while T is based on the network configuration. In our framework, the decomposition-based prediction model is extremely lightweight, so any additional cost associated with running our system is assumed to be negligible. For more advanced models, such as LSTMs, this cost could become non-negligible and would need to be explicitly accounted for in the overall utility computation.

6.3.2 Dissecting Energy Saving Gains

Deploying our solution at scale in the wild is impractical due to operational costs and the difficulty of reproducing identical network conditions across different scenarios. Instead, we perform a *what-if* analysis on our collected dataset to quantify the achievable gains under idealized best- and worst-case scenarios. To properly quantify this potential saving, we conducted controlled experiments that allow us to keep identical network conditions in different scenarios. Thanks to the repeatability, we can verify and perform statistical analysis of the results to strengthen our estimates. Due to uncertainty in the detailed operation of the modem chipset, in this paper, we report the range of possible savings, which we call the best- and worst-case savings.

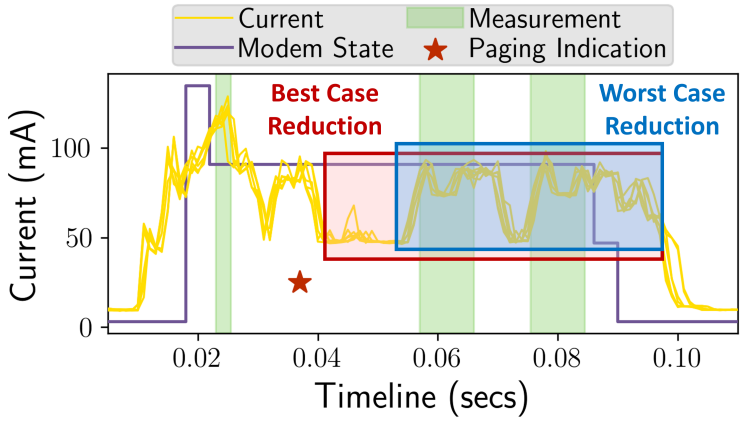


Figure 6.9: PO duration and energy-saving bounds for inter-frequency measurement relaxation.

Figure 6.9 illustrates our approach for estimating bounds on energy savings for relaxing inter-frequency measurements on bands N25 and N41. The *best case reduction* assumes the modem enters its sleep state immediately after decoding the paging message, whereas the *worst case reduction* assumes it sleeps only after the point in time when the first inter-frequency measurement would have been conducted. Both cases incorporate ramp-up and ramp-down overheads, i.e., the short power spikes before the modem is fully active and after it transitions to sleep. PO duration is measured from the time the power rises above the idle-mode floor until it returns to that floor.

Table 6.2: Best- and worst-case inter-frequency relaxation scenarios for N25 and N41 bands.

Band		**N25**	**N41**
Overall PO	Duration (ms)	93.0	48.0
	Current (mA)	6.5	3.6
Best Case Reduction	Duration (ms)	52.9	17.0
	Current (mA)	3.8	1.2
Worst Case Reduction	Duration (ms)	42.7	9.0
	Current (mA)	3.2	0.6

Table 6.2 summarizes, for each band, the average PO duration, current consumption, and the duration and current reductions achieved in the best- and worst-case scenarios. These values form the basis for our savings estimates.

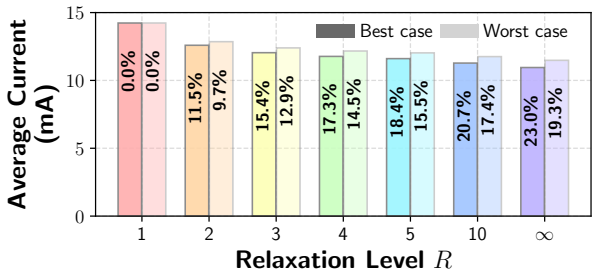


Figure 6.10: Estimated best- and worst-case energy savings for N25 across relaxation factors R. $R = \infty$ represents the upper bound where all inter-frequency measurements are removed.

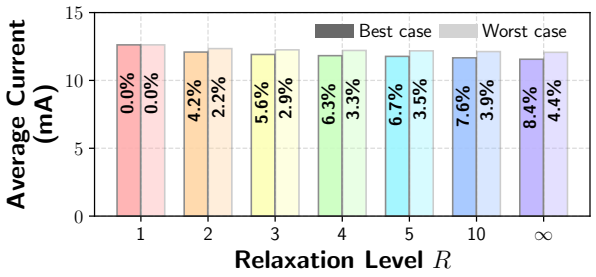


Figure 6.11: Estimated best- and worst-case energy savings for N41 across relaxation factors R, showing smaller gains than N25 due to shorter paging occasions and fewer inter-frequency measurements.

Figure 6.10 compares best- and worst-case energy savings for N25 as a function of the relaxation factor R. Percentage savings are shown relative to the baseline case ($R = 1$, no relaxation). The $R = \infty$ case—where all inter-frequency measurements are removed—represents the empirical upper bound. Energy savings exhibit diminishing returns: for example, $R = 2$ captures roughly half of the maximum possible savings, which translates to 9.7-11.5% idle energy savings. This means that even conservative relaxation factors like $R = 2$ can offer significant gains. Similarly, $R = 10$ achieves 17.4–20.7% savings, which is within 1.9–2.3% of the upper bound ($R = \infty$ case).

For N41 (Figure 6.11), savings are significantly smaller due to shorter PO durations and only one inter-frequency measurement per PO. At $R = 10$, best- and worst-case savings are only 3.9–7.6%, compared to 17.4–20.7% for N25.

This analysis highlights that achievable savings vary widely by the different measurement configurations applied to different bands. In scenarios where potential gains are minimal (e.g., under medium-to-high mobility), the overhead of running a complex adaptive relaxation mechanism may outweigh the benefits. These results motivate incorporating *energy-gain awareness* into

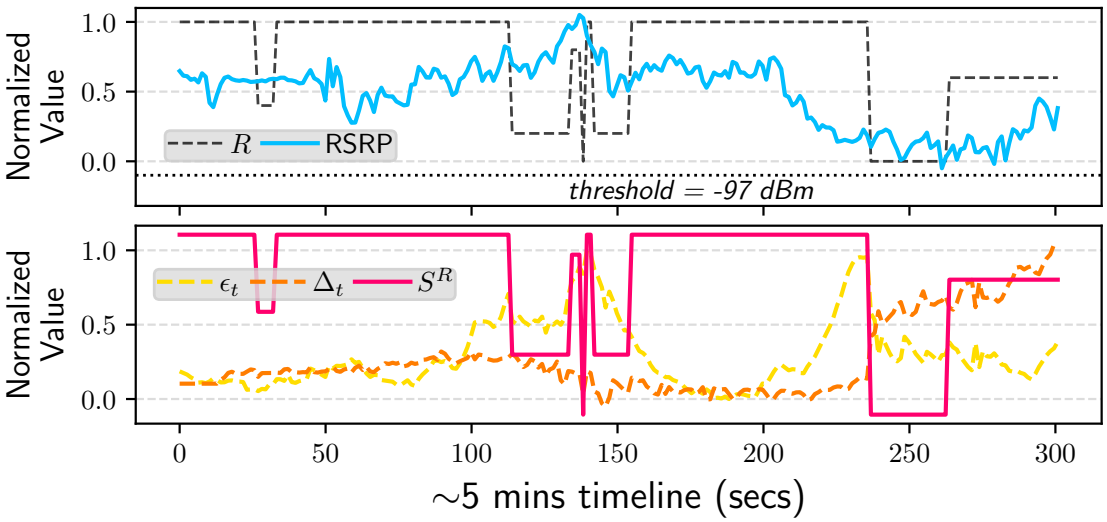


Figure 6.12: Adaptive relaxation dynamics under a low-mobility walking trace.

the relaxation decision process, ensuring that adaptation is only triggered when savings justify the operational cost.

6.3.3 End-to-end Results

We now evaluate our adaptive relaxation framework in an end-to-end setting by replaying measurement traces through the simulator. The system computes relaxation cost (C_t), saving (S^R), and overall utility (U_t^R) for each paging occasion, balancing energy efficiency against reselection risk. Our experiments use the Full Search Strategy to find the best R.

Figure 6.12 illustrates the temporal dynamics of our adaptive relaxation scheme under a representative low-mobility trace, where the user is walking with the phone in hand. At the beginning of the trace, the signal quality is well above the reselection threshold, resulting in a high relaxation factor ($R = 10$). This reduces the frequency of measurements without incurring reselection risk.

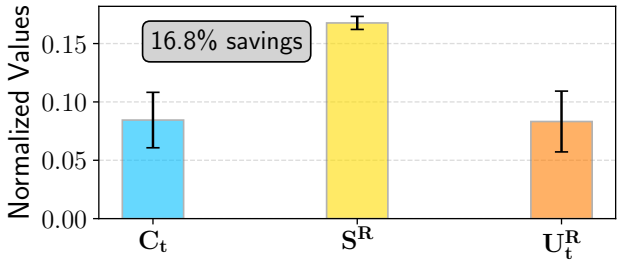


Figure 6.13: Overall utility (U_t^R) across the full trace.

In the middle of the trace, the signal quality suddenly improves, producing a sharp increase in the prediction error ϵ_t. Since the system had been operating with a high relaxation level ($R = 10$, i.e., one measurement every 12.8 secs), insufficient recent measurements were available to maintain accurate prediction. The adaptive policy reacts by momentarily lowering R, collecting additional data to stabilize the predictor, and then increasing relaxation again after some fluctuation. This behavior reflects the intrinsic trade-off: aggressive skipping amplifies prediction uncertainty, which our solution self-corrects by temporarily reducing relaxation.

Towards the end of the trace, the signal quality drops close to the reselection threshold. The distance-to-threshold penalty Δ_t rises sharply, and the policy promptly reduces R to 1, ensuring responsiveness to potential reselection events. After the channel recovers, relaxation resumes, but conservatively (e.g., $R = 7$ instead of $R = 10$), reflecting the system's caution after recent instability.

Overall, this result demonstrates that the proposed scheme adapts its behavior intelligently to channel dynamics: it exploits good conditions for energy savings, corrects for prediction uncertainty when data becomes sparse, and aggressively reduces relaxation when reselection risk increases. This adaptive modulation of R highlights the robustness of the approach compared to static relaxation strategies.

Figure 6.13 summarizes the results by reporting the mean and standard deviation of C_t, S^R, and U_t^R across the full trace. On average, the proposed scheme achieves **16.8% energy savings**, with utility remaining consistently positive despite variability in channel conditions. The cost values remain moderate, indicating that the adaptive policy avoids aggressive relaxation when reselection risk is high.

These results demonstrate that meaningful idle-mode power reductions can be achieved without compromising mobility responsiveness. It highlights that predictive adaptation can provide robust energy savings, compared to fixed relaxation heuristics.

6.3.4 Micro-benchmarking

We further benchmark the prediction models that drive our adaptive relaxation policy against popular machine learning baselines for channel prediction, including a 1D CNN (CNN1D), an LSTM, and a linear regression model. The input consists of signal quality (RSRP) traces with history length $k = 8$. To evaluate performance under different relaxation levels R (2, 4, and 10), we down-sample the traces accordingly and train and test each model separately for every R value, ensuring that both training and evaluation reflect the reduced measurement availability. We show performance in terms of mean absolute error (MAE), root mean squared error (RMSE), and correlation with ground truth. All models are implemented in PyTorch and trained on the same dataset split described earlier (70% train, 30% test).

Table 6.3 summarizes the results. Across all relaxation levels, our proposed decomposition-based model, which explicitly separates RSRP into trend and residual components, consistently achieves the best performance, with MAE as low as 0.50 and correlation above 0.91 at $R = 2$. In

Table 6.3: Prediction performance of different models across relaxation levels. Each cell reports MAE / RMSE / Correlation, with lower error and higher correlation indicating better performance.

Model	**# of learnable params**	$\mathbf{R = 2}$	$\mathbf{R = 4}$	$\mathbf{R = 10}$
CNN1D	161	0.78 / 1.22 / 0.87	0.86 / 1.32 / 0.81	1.84 / 2.16 / 0.45
Ours	18	0.50 / 0.94 / 0.92	0.71 / 1.21 / 0.85	1.26 / 1.65 / 0.63
LSTM	17217	0.68 / 1.14 / 0.88	1.09 / 1.46 / 0.82	1.51 / 1.85 / 0.56
Linear	9	2.97 / 3.39 / 0.48	1.96 / 2.40 / 0.58	2.77 / 3.42 / 0.13

comparison, CNN1D and LSTM provide reasonable accuracy at low R but degrade noticeably with higher relaxation levels (e.g., CNN1D MAE rises to 1.84 at $R = 10$). The vanilla linear regression model performs poorly, particularly at $R = 10$, highlighting its inability to capture temporal dependencies in the signal when input data is coarse. Moreover, unlike the other methods, it fails to exhibit any consistent trend as R increases, suggesting that the model is too simplistic to effectively learn the underlying signal dynamics.

A key advantage of our method is its efficiency. Despite outperforming the more complex CNN and LSTM models, it remains lightweight, with only 18 learnable parameters. This small footprint makes it suitable for deployment on power-constrained UEs, while still delivering high predictive accuracy across a wide range of relaxation levels. Overall, these results demonstrate that our design offers a robust and practical solution for adaptive measurement relaxation in 5G and future 6G systems.

6.4 Conclusion and Future Directions

This work presents the design and evaluation of a complete prediction-driven framework for idle-mode measurement relaxation. Our results demonstrate that adaptive relaxation can reduce idle-mode energy consumption by up to 20%, highlighting the effectiveness of prediction-guided and

context-aware measurement control. Beyond the algorithmic contribution, we provide detailed power measurements from a commercial 5G network (T-Mobile) and an off-the-shelf smartphone (Samsung Galaxy S24+), offering hardware-grounded evidence of real-world energy trade-offs. Such measurement-driven validation remains limited in existing literature, yet is essential for understanding the practical impact of energy-saving mechanisms.

While the results are encouraging, several aspects warrant further investigation. First, our experiments primarily focus on stationary and low-mobility scenarios. Extending the analysis to medium- and high-mobility users, rural deployments, and denser urban environments would provide a more complete characterization of the framework's robustness under diverse channel dynamics. Second, the effectiveness of the proposed method depends on system parameters such as the history buffer length k and weight factors (λ, η, ϕ). Although empirically chosen values performed well in our study, future work should examine adaptive parameter tuning or derive operating-point guidelines that generalize across network conditions.

Finally, our evaluation was conducted on a single commercial device. Different hardware platforms may exhibit distinct power behaviors due to modem design, RF frontend characteristics, or power management implementations. In particular, emerging device categories such as reduced-capability (RedCap) UEs, which operate under tighter energy constraints, may benefit disproportionately from adaptive measurement relaxation. Systematic cross-device evaluation therefore represents a valuable next step.

In summary, this work establishes the feasibility of prediction-driven idle-mode measurement control and demonstrates its tangible energy-saving potential. By formalizing the trade-off between energy reduction and mobility reliability, it lays a foundation for integrating adaptive measurement policies into future 5G and beyond networks.

Chapter 7

Related Work

Related work spans several complementary areas — mobility management and handover prediction, duplex/TDD resource allocation, connectivity (cell selection/CA) and profiling, application-aware QoE control, and UE idle-mode energy optimization. In the sections that follow, we briefly summarize representative efforts in each area and contrast them with our contributions, emphasizing where our frameworks and systems extend prior art by unifying performance-driven control, operator-aware policies, and prediction-guided, hardware-validated energy savings.

7.1 Mobility Management and Handover Prediction

Cellular handover and mobility measurement. Prior work has extensively characterized mobility management behavior in operational cellular networks. Several studies [50, 95, 184] analyze real-world handover configurations and highlight the complexity of carrier-deployed mobility policies. Early measurement studies were conducted primarily in 3G and LTE networks [95, 50, 91]. Li *et al.* [95] identify persistent handover loops and propose methods to detect mobility instability, while Deng *et al.* [50] perform large-scale handover analysis across LTE, 3G, and 2G

networks, demonstrating significant diversity in handover parameter configurations. Similarly, Li *et al.* [91] analyze signaling traces collected using MobileInsight [94] to reveal instability in operational mobility procedures.

More recent work has shifted toward understanding 5G deployments. Measurement studies [182, 116, 117, 122, 121] characterize throughput, beam management, and cross-RAT interactions in commercial 5G networks. Xu *et al.* [182] focus on sub-6 GHz deployments and show that handovers significantly impact signal quality and TCP throughput. Narayanan *et al.* [121] provide a measurement-driven analysis of mmWave deployments, highlighting the effects of beam switching and coverage fragmentation. In contrast to these works, our study provides a comprehensive characterization of 5G mobility management across multiple bands, deployment modes, and application workloads, while also quantifying handover impacts on energy efficiency and application QoE.

Handover prediction. Predicting handovers to proactively adapt to network dynamics has been explored in prior generations of cellular systems. Early approaches for 3G and LTE leveraged mobility trajectories or historical signal trends to anticipate future handovers [84, 64]. Later work exploited temporal correlation and location-based features for prediction [130]. More recently, Mei *et al.* [112] employ gradient-boosting models using lower-layer measurements to predict handover events. However, these approaches primarily focus on prediction accuracy in isolation and do not evaluate the practical utility of handover prediction for improving application performance.

In contrast, our work develops a holistic handover prediction framework that models both measurement evolution and base-station decision logic, enabling explainable and early handover

predictions. Furthermore, we evaluate the system using real applications as case studies, demonstrating measurable QoE improvements under mobility.

7.2 Duplex Resource Allocation

Dynamic TDD schemes. Several prior works propose dynamic TDD schemes for LTE/5G to adapt UL/DL slot allocation to traffic or interference conditions [29, 37, 36, 132, 161, 164, 169, 52, 190]. The closest to our approach is DRP [29], which uses deep reinforcement learning and per-UE buffer reports (BSRs) to derive TDD decisions. Compared to DRP, Wixor is designed with a heavier emphasis on practical deployability: DRP (i) largely ignores the space–time dynamics of wireless channels, (ii) does not explicitly account for diverse QoS/QoE needs across users and flows, (iii) relies on per-UE features which limits scalability as the number of active UEs grows, and (iv) does not treat several important protocol-level details such as S&S arrangement and guard-period overhead. Wixor addresses these gaps by using compact BS-level feature aggregates, explicitly modeling guard periods and inter-slot delay, and optimizing a QoS-aware reward that can be tuned for latency- or throughput-sensitive workloads. We experimentally compare with DRP in §4.7.

Other dynamic-TDD literature tackles specific scenarios (e.g., high-mobility HetNets [164], dense deployments [169], massive IoT [132], and small cells or D2D-assisted designs [52, 190, 161]). Those works are valuable when applied to the niche settings they target, but they do not offer the end-to-end, application-transparent, deployable policy derivation and slot-arrangement pipeline that Wixor provides for both public and private 5G deployments.

5G/LTE resource scheduling. MAC-layer resource allocation and scheduling (e.g., proportional-fair, slice-aware, or latency-aware schedulers) is a large literature stream [43, 42, 20, 108, 187]. Examples include RadioSaber and iRSS for network slicing contexts [42, 187], SMART for massive MIMO settings [20], and ELASE / UQ-vRAN for vRAN deployments [43, 108]. These works focus on *how* the BS allocates symbols/resources given a TDD pattern and active UE set; by contrast, Wixor addresses a complementary problem: *which* TDD pattern (slot percentage and symbol arrangement) the BS should adopt to meet QoS objectives. In practice the two problems are orthogonal — Wixor's policy derivation can operate on top of or alongside advanced MAC schedulers.

Application-specific optimizations in cellular networks. A number of systems optimize particular classes of applications over cellular/WiFi links, e.g., low-latency video analytics, cloud gaming, and interactive video conferencing [26, 83, 113, 147, 163, 181]. Representative systems such as DChannel, Tutti, LRP, and Zhuge optimize transport/application-layer behavior by leveraging application-level QoE signals or cross-layer feedback [147, 181, 163, 113]. Those approaches are effective when application cooperation is available, but they require explicit application-side instrumentation and typically target a single application class. Wixor takes a different, application-agnostic path: it is *application-transparent* and relies on radio-protocol-layer QoS indicators (BS-level aggregates) as proxies for application performance. That design lets Wixor simultaneously improve a heterogeneous mix of applications without requiring per-application QoE feedback or changes to end-host stacks.

Our work sits at the intersection of dynamic-TDD design, RL-based control for RANs, and application-aware networking. Unlike prior DRL-based TDD schedulers that use per-UE inputs and ignore protocol-level constraints, and unlike application-specific optimizers that require app

cooperation, Wixor combines (i) a compact, transferable BS-level feature design, (ii) an RL-based demand predictor trained in a trace-driven simulator, and (iii) a practical policy-provision stage that explicitly handles guard periods and slot arrangement — yielding a system that is both practical to deploy and effective across a broad set of real-world workloads (see §4.7).

7.3 Connectivity Management

Cell Re/selection and Handovers. Prior work has explored many facets of cell selection, reselection, and handover to improve access speed and user throughput. Examples include device-side assistance and network-side parameter tuning to favor higher throughput or more stable connections [49, 89, 50]. iCellSpeed [49] adopts a proactive, device-side approach that helps UEs select cells that improve throughput; Li *et al.* [89] show how reconfiguring operator-defined selection parameters can dynamically improve throughput. A range of measurement and analysis studies highlight practical issues such as persistent handover loops, instability, and suboptimal access decisions in deployed 4G/5G networks [72, 91, 135, 192, 102] and propose remedies including automated detection and parameter adjustment.

Our work differs from these efforts in three respects. First, rather than optimizing a single CM procedure or a single metric (e.g., throughput), OPCM offers a unified abstraction that decouples performance objectives (throughput, latency, energy, etc.) from the underlying CM actions. Second, unlike many UE-side methods, OPCM is a centralized BS-side framework designed to enforce operator policies (fairness, load balancing) across multiple UEs. Third, OPCM reduces profiling overhead by combining lightweight active exploration with passive, correlation-based performance approximation, enabling practical multi-UE decisions at scale.

Carrier Aggregation and Multi-Carrier Management. Carrier Aggregation (CA) and related multi-carrier mechanisms are central to boosting capacity in modern networks. Recent deployment analyses and measurement studies investigate how CA is used in real networks and the operational tradeoffs it introduces [189, 92, 87]. Work on CA reliability and failure modes explores the risks in adding secondary carriers and handling access failures [101, 103]. CA++ and related algorithmic efforts propose enhancements to CA behavior such as group-based activation or smarter carrier selection [90]. Our measurements (see §5.4.1) show operators often engage in group-based CA in practice, which motivates pruning combinatorial carrier options in OPCM and aligns our design with operational realities. OPCM is complementary to CA optimization work and can be integrated with advanced CA/aggregation mechanisms to further improve user-centric performance.

Measurement Studies and Profiling. A large body of empirical work characterizes 5G performance, energy consumption, and mobility behavior across devices, operators, and environments [122, 182, 120, 118, 188, 72]. These studies reveal important patterns—e.g., the effect of reusing legacy bands for 5G, energy–throughput tradeoffs across device models, and mobility-related overheads—that motivate practical system designs. Prior measurement efforts also inform techniques for estimating cross-band and cross-cell performance [93, 172, 30]. OPCM builds on these insights by (i) characterizing the wide availability and heterogeneity of multi-cell deployments, (ii) measuring the temporal stability and performance diversity of cell combinations, and (iii) leveraging time-lagged cross-correlation to passively approximate inactive combinations' performance rather than relying exclusively on active probing.

Application-aware and QoE-driven Network Control. Recent systems seek to optimize application-level QoE over wireless links, for example by adapting network or application behavior for video streaming, AR/VR, or edge inference workloads [26, 83, 113, 147, 163, 181]. These approaches typically require application feedback or client-side instrumentation to achieve fine-grained QoE objectives. In contrast, OPCM is application-transparent: it infers workload preferences via standardized indicators (e.g., 5QI) and BS-observable metrics, enabling operator-controlled, cross-user QoE optimization without modifying the applications or requiring client cooperation.

RL and Learning-based RRM. Machine learning and reinforcement learning have been applied to radio resource management, dynamic TDD adaptation, and handover optimization [29, 42, 20, 108]. RL-based schemes often operate on per-UE features and can be powerful when rich labeled data and careful training are available. However, RL solutions may face practical challenges in operator networks—training stability, per-UE state explosion, and policy compliance with operator-level objectives. OPCM takes a pragmatic stance: it uses lightweight estimation, objective-aware pruning, and opportunistic exploration to achieve robust, policy-compliant CM decisions that are practical for deployment.

In summary, OPCM draws from and complements a broad literature spanning mobility management, carrier aggregation, empirical measurements, application-aware control, and ML-driven RRM. Its novelty lies in unifying these ideas into a practical, operator-friendly framework that (i) decouples performance objectives from CM procedures, (ii) enforces RAN policies at the BS, and (iii) uses hybrid profiling (active + passive, correlation-aware) to scale multi-UE CM decisions with modest overheads. Where possible, we compare against representative baselines (e.g., legacy CM and iCellSpeed) in §5.8 to quantify these advantages.

7.4 Idle-State Measurement Adaptation

We review prior work across four areas: 3GPP measurement and energy-saving mechanisms, empirical UE power characterization, prediction-driven mobility control, and optimization of reselection and handover procedures.

3GPP Measurement and Energy-Saving Mechanisms. 3GPP specifications define idle- and inactive-mode Radio Resource Management (RRM) measurements, including intra- and inter-frequency scanning for cell reselection and mobility. In contrast to connected mode, where measurement gaps are coordinated with active data sessions, idle-mode measurements occur periodically and can incur non-trivial energy overhead. To mitigate this cost, TS 38.304 [10] specifies relaxation triggers such as *low mobility* and *non–cell-edge* conditions. Industry reports extend these mechanisms. MediaTek [110] discusses relaxation for Radio Link Management (RLM) and Beam Failure Detection (BFD), while comparative studies evaluate DRX, wake-up signals, and RRM relaxation trade-offs [33]. Broader surveys [86] cover O-RAN and adaptive carrier activation strategies, and forward-looking proposals such as LP-WUS for 6G [144] suggest continued evolution of measurement control. However, these efforts largely rely on static or rule-based triggers rather than predictive adaptation.

Empirical Studies of UE Power Consumption. Several measurement-driven studies quantify the energy impact of RRM procedures. Xu *et al.* [183] demonstrate that 5G DRX and idle-mode behavior can significantly increase UE power consumption relative to LTE. MediaTek [111] evaluates Release 17 relaxation gains, while Chabi [39] reports over 20% idle power reduction through optimized NR measurements. Large-scale field studies [122, 73] further reveal persistent RRM activity even in stable channel conditions, highlighting untapped opportunities for adaptive

measurement control. While these works quantify energy impact, they do not propose predictive mechanisms to modulate measurement frequency based on channel evolution.

Prediction-Driven Mobility Control. Prediction-based approaches have been explored primarily for connected-mode mobility optimization. Panitsas *et al.* [131] apply deep learning to forecast handovers for performance and energy gains. Hybrid models combining statistical learning and ML have also been proposed: Kaur *et al.* [82] use LSTM and SVM for proactive handover triggering, and Wang *et al.* [176] mitigate redundant handovers in dense deployments. Reinforcement learning formulations [186] treat mobility control as a contextual decision problem. These studies focus primarily on active-mode handover management, whereas idle-mode measurement scheduling remains comparatively underexplored.

Energy-Aware Reselection and Mobility Optimization. Energy-aware mobility optimization has been considered in both idle and connected states. Elbatal *et al.* [55] adapt handovers based on traffic and channel conditions, and AI-driven RAN optimization frameworks are surveyed in [59]. While these approaches demonstrate the feasibility of intelligent mobility control, they do not explicitly address the trade-off between idle-mode measurement frequency and reselection reliability.

Unlike prior work that relies on static relaxation triggers or focuses on connected-mode optimization, our work targets idle-mode measurement relaxation using a predictive, adaptive framework. By explicitly modeling prediction error and reselection risk, we formalize the trade-off between energy savings and mobility responsiveness, moving beyond threshold-based heuristics. We also validate our design with hardware-grounded power measurements using both external monitors and modem-level SDM logs, providing concrete evidence of achievable energy savings in operational 5G deployments.

Chapter 8

Conclusion and Future Work

This dissertation examined how core resource management mechanisms in modern cellular networks can be improved through *measurement-driven, cross-layer decision-making*. Although contemporary 5G networks expose rich observability across the PHY, MAC, RRC, transport, and application layers, many operational mechanisms—including mobility management, duplex resource allocation, connectivity management, and idle-mode measurement scheduling—remain governed by static heuristics and threshold-based logic. While these mechanisms maintain link reliability and coverage, they often fail to optimize system-level objectives such as application QoE, network QoS, fairness, and UE energy efficiency.

The central thesis of this work is that incorporating prediction, cross-layer signals, and system-level objectives into control-plane decision-making can significantly improve performance while remaining compatible with existing 3GPP procedures. To validate this, this dissertation combines large-scale empirical measurement studies with the design and evaluation of practical systems across four domains in operational 5G networks: mobility management, duplex resource allocation, connectivity management, and idle-mode measurement scheduling.

Through extensive measurements, system design, and prototype evaluation, the dissertation demonstrates that incorporating prediction and cross-layer signals into cellular control loops can substantially improve application performance and energy efficiency without requiring modifications to standardized cellular protocols. The remainder of this chapter summarizes the main contributions of this work, outlines promising directions for future research, and concludes with broader implications for next-generation cellular systems.

8.1 Summary of Contributions

• This dissertation presents one of the most comprehensive empirical characterizations of mobility behavior in operational 5G networks. Using a large-scale measurement campaign spanning more than 6,200 km and over 47,000 mobility events, we quantify the frequency, performance impact, and energy implications of handovers across diverse bands and deployment architectures.

• Building on these insights, we design Prognos, a predictive mobility framework that anticipates handover events by modeling both signal evolution and base-station decision logic. Prognos enables proactive mitigation of handover disruptions and improves application QoE under mobility.

• This work investigates duplex resource allocation in 5G Time Division Duplex (TDD) networks and shows that widely deployed static or semi-static TDD configurations are misaligned with modern asymmetric workloads, leading to uplink under-provisioning and latency inflation.

• To address this limitation, we design Wixor, a predictive dynamic TDD adaptation system that jointly optimizes UL/DL slot distribution and arrangement. Wixor leverages cross-layer base-station features and reinforcement learning to derive QoS-aware TDD policies while respecting practical protocol constraints.

• Through large-scale measurements across multiple cities and operators, we show that heterogeneous multi-cell deployments expose substantial performance diversity across feasible cell combinations, yet legacy connectivity management mechanisms remain primarily signal-driven.

• To exploit this diversity, we design OPCM, a centralized performance-driven connectivity management framework that prunes feasible cell combinations, profiles their performance using hybrid measurement techniques, and opportunistically selects configurations that improve throughput, latency, and energy efficiency while respecting RAN policies.

• This dissertation further investigates idle-mode measurement scheduling and shows that periodic RRM measurements can contribute significant UE energy overhead, particularly due to repeated inter-frequency scans under stable channel conditions.

• We develop PARMA, a predictive adaptive measurement relaxation framework that dynamically adjusts measurement frequency using signal prediction, reselection risk modeling, and lightweight energy estimation. PARMA reduces idle-mode energy consumption while preserving reliable cell reselection behavior.

• Collectively, these contributions demonstrate that diverse cellular resource management mechanisms can be systematically structured as measurement-driven, cross-layer closed-loop control systems that explicitly optimize system-level objectives while remaining compatible with existing 3GPP procedures.

8.2 Future Research Directions

While this dissertation demonstrates the benefits of measurement-driven control across several resource management domains, many opportunities remain for further research.

Unified cross-layer control frameworks. Each system presented in this dissertation targets a specific control surface in the cellular stack. However, these mechanisms are inherently interconnected. For example, mobility events influence duplex demand, connectivity decisions affect scheduling performance, and measurement activity impacts energy consumption. Future work could explore unified frameworks that coordinate multiple control loops simultaneously, enabling joint optimization across mobility management, duplex configuration, connectivity management, and energy-aware operation.

Integration with emerging RAN architectures. The increasing programmability of cellular infrastructure—through software-defined base stations, open RAN architectures, and radio intelligent controllers (RICs)—creates new opportunities for deploying measurement-driven control mechanisms. Future systems could integrate the designs proposed in this dissertation with emerging RAN platforms, enabling operators to deploy predictive resource management policies across distributed edge nodes and centralized control planes.

Learning-driven cellular control. Machine learning and artificial intelligence are expected to play an increasing role in future cellular systems. Although this dissertation uses learning selectively for prediction tasks, broader opportunities exist for integrating adaptive learning models into network control loops. Future work could explore models that continuously learn from live network data, incorporate contextual information such as mobility patterns or application workloads, and dynamically refine control policies as network conditions evolve.

Application–network cooperation. Another promising direction is the development of standardized interfaces that allow applications to communicate performance objectives to the network. Emerging mechanisms such as network exposure functions and radio network information

services could enable applications to adapt behavior based on network predictions or to inform the network of QoS requirements.

Next-generation wireless systems The ideas explored in this dissertation are likely to become even more relevant in future wireless systems. Upcoming 5G-Advanced and 6G networks are expected to incorporate greater heterogeneity in spectrum, deployment density, and service requirements. Measurement-driven, predictive control frameworks such as those developed in this dissertation provide a promising foundation for designing such systems.

8.3 Concluding Remarks

Cellular networks have undergone remarkable technological evolution, yet many operational mechanisms remain rooted in rule-based designs developed for earlier generations. As networks grow increasingly complex and application demands become more diverse, static heuristics are no longer sufficient to achieve optimal performance. This dissertation shows that several core cellular resource management functions can be systematically interpreted as *measurement-driven, cross-layer control problems*. By incorporating cross-layer observations, predictive modeling, and explicit system-level objectives into the decision loop, it becomes possible to improve application QoE, network performance, and UE energy efficiency while remaining compatible with existing 3GPP procedures as well as operator policies.

Through empirical measurements and practical system designs, this work demonstrates how predictive intelligence can transform mobility management, duplex resource allocation, connectivity management, and idle-mode measurement scheduling. Rather than proposing isolated optimizations, the dissertation introduces a general methodology for designing intelligent control

mechanisms in cellular networks. As cellular systems continue to evolve toward increasingly programmable and data-driven architectures, measurement-driven cross-layer control is likely to become a key principle guiding the design of next-generation wireless networks.

Bibliography

[1] 3GPP. *NR; NR and NG-RAN Overall Description; Stage-2*. Technical Specification (TS) TS 38.300. Version 17.0.0. 3rd Generation Partnership Project (3GPP), 2022.

[2] 3GPP. *NR; Physical channels and modulation*. Technical Specification (TS) TS 38.211. Version 17.0.0. 3rd Generation Partnership Project (3GPP), 2022.

[3] 3GPP. *NR; Physical layer procedures for control*. Technical Specification (TS) TS 38.213. Version 17.0.0. 3rd Generation Partnership Project (3GPP), 2022.

[4] 3GPP. *NR; User Equipment (UE) radio access capabilities*. Tech. rep. TS 38.306. Version 17.1.0. Release 17. 3rd Generation Partnership Project (3GPP), 2023. URL: https://www.etsi.org/deliver/etsi_ts/138300_138399/138306/17.01.00_60/ts_138306v170100p.pdf.

[5] *3GPP TS 36.331: Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (V16.3.0)*. 2021.

[6] *3GPP TS 37.340: NR; Multi-connectivity; Overall description; Stage-2 (V15.3.0)*. 2019.

[7] *3GPP TS 38.133 V16.4.0: Requirements for Support of Radio Resource Management*. Release 16. 2020. URL: https://www.etsi.org/deliver/etsi_ts/138100_138199/138133/16.04.00_60/ts_138133v160400p.pdf.

[8] *3GPP TS 38.211: Physical channels and modulation (V15.8.0)*. 2020.

[9] *3GPP TS 38.215: 5G NR: Physical layer measurements (V16.2.0)*. 2020.

[10] *3GPP TS 38.304 V17.3.0: NR; User Equipment (UE) Procedures in Idle Mode and in RRC Inactive State*. Release 17. 2022. URL: https://www.3gpp.org/ftp/Specs/archive/38_series/38.304/.

[11] *3GPP TS 38.321: Medium Access Control (MAC) protocol specification (V15.6.0)*. 2019.

[12] *3GPP TS 38.331: NR; Radio Resource Control (RRC); Protocol specification (V16.3.1)*. 2021.

[13] *3GPP TS 38.912: Study on New Radio (NR) access technology (V15.0.0)*. 2018.

[14] *5G MAC BSR – Buffer Status Reporting*. 2024. URL: https://www.techplayon.com/5g-mac-bsr-buffer-status-reporting/.

[15] *5G NR Frequency Bands*. 2024. URL: https://en.wikipedia.org/wiki/5G_NR_frequency_bands.

[16] *5G-LENA: ns-3 module to simulate 5G NR networks*. 2024. URL: https://apps.nsnam.org/app/nr/.

[17] A. A. M. K. Abuelgasim and K. M. Yusof. "High Speed Mobility Management Performance in a Real LTE Scenario". In: *Engineering, Technology amp; Applied Science Research* 10 (2020), pp. 5175–5179. DOI: 10.48084/etasr.3245.

[18] *Accuver XCAL*. 2022. URL: https://www.accuver.com/sub/products/view.php?idx=6.

[19] *Add 5G capabilities to your app*. 2022. URL: https://developer.android.com/about/versions/11/features/5g.

[20] Qing An, Santiago Segarra, Chris Dick, Ashutosh Sabharwal, and Rahman Doost-Mohammady. "A Deep Reinforcement Learning-Based Resource Scheduler for Massive MIMO Networks". In: *IEEE Transactions on Machine Learning in Communications and Networking* (2023).

[21] *An end-to-end platform for machine learning*. 2024. URL: https://www.tensorflow.org/.

[22] Android Developers. *Power Profiler - Android Studio*. https://developer.android.com/studio/profile/power-profiler. Accessed: 2025-05-28. 2024.

[23] *Ant Media: liveVideoBroadcaster*. 2024. URL: https://github.com/ant-media/LiveVideoBroadcaster.

[24] Guilherme H Apostolo, Pablo Bauszat, Vinod Nigade, Henri E Bal, and Lin Wang. "Live video analytics as a service". In: *Proceedings of the 2nd European Workshop on Machine Learning and Systems*. 2022, pp. 37–44.

[25] Kelvin Au, Liqing Zhang, Hosein Nikopour, Eric Yi, Alireza Bayesteh, Usa Vilaipornsawai, Jianglei Ma, and Peiying Zhu. "Uplink contention based SCMA for 5G radio access". In: *2014 IEEE Globecom Workshops (GC Wkshps)*. 2014.

[26] Jose A. Ayala-Romero, Andres Garcia-Saavedra, Xavier Costa-Perez, and George Iosifidis. "EdgeBOL: automating energy-savings for mobile edge AI". In: *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*. CoNEXT '21. 2021.

[27] Duin Baek, Mallesham Dasari, Samir R Das, and Jihoon Ryoo. “dcSR: practical video quality enhancement using data-centric super resolution”. In: *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*. 2021, pp. 336–343.

[28] Carlos Baena, Sergio Fortes, O. S. Peñaherrera-Pulla, Eduardo Baena, and Raquel Barco. “Gaming in the Cloud: 5G as the Pillar for Future Gaming Approaches”. In: *IEEE Communications Magazine* 62.11 (2024), pp. 76–82. DOI: 10.1109/MCOM.005.2300256.

[29] Miloud Bagaa, Karim Boutiba, and Adlen Ksentini. “On using Deep Reinforcement Learning to dynamically derive 5G New Radio TDD pattern”. In: *2021 IEEE Global Communications Conference (GLOBECOM)*. 2021.

[30] Arjun Bakshi, Yifan Mao, Kannan Srinivasan, and Srinivasan Parthasarathy. “Fast and Efficient Cross Band Channel Prediction Using Machine Learning”. In: *The 25th Annual International Conference on Mobile Computing and Networking*. MobiCom ’19. Los Cabos, Mexico: Association for Computing Machinery, 2019. ISBN: 9781450361699. DOI: 10.1145/3300061.3345438.

[31] Giovanni Bartolomeo, Jacky Cao, Xiang Su, and Nitinder Mohan. “Characterizing distributed mobile augmented reality applications at the edge”. In: *Companion of the 19th International Conference on emerging Networking EXperiments and Technologies*. 2023, pp. 9–18.

[32] Gilberto Berardinelli, Klaus I. Pedersen, Frank Frederiksen, and Preben Mogensen. “On the Guard Period Design in 5G TDD Wide Area”. In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. 2016, pp. 1–5. DOI: 10.1109/VTCSpring.2016.7504377.

[33] Vitalii Beschastnyi, Darya Ostrikova, Dmitri Moltchanov, Yuliya Gaidamaka, Yevgeni Koucheryavy, and Konstantin Samouylov. “Comparison of energy conservation strategies for 5G NR RedCap service in industrial environment”. In: *Computer Networks* 254 (2024). DOI: 10.1016/j.comnet.2024.110792.

[34] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “Yolov4: Optimal speed and accuracy of object detection”. In: *arXiv preprint arXiv:2004.10934* (2020).

[35] Leonardo Bonati, Michele Polese, Salvatore D’Oro, Stefano Basagni, and Tommaso Melodia. “Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead”. In: *Computer Networks* (2020).

[36] Karim Boutiba, Miloud Bagaa, and Adlen Ksentini. “Multi-Agent Deep Reinforcement Learning to Enable Dynamic TDD in a Multi-Cell Environment”. In: *IEEE Transactions on Mobile Computing* (2024).

[37] Karim Boutiba, Miloud Bagaa, and Adlen Ksentini. “On enabling 5G Dynamic TDD by leveraging Deep Reinforcement Learning and O-RAN”. In: *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. 2023.

[38] *Brawlhalla*. 2022. URL: https://store.steampowered.com/app/291550/Brawlhalla/.

[39] A. F. Chabi et al. “Power Consumption Evaluation Using 5G Energy Saving Features”. In: *Proc. IEEE ICC*. 2024.

[40] Hyunseok Chang, Matteo Varvello, Fang Hao, and Sarit Mukherjee. “Can You See Me Now? A Measurement Study of Zoom, Webex, and Meet”. In: *Proceedings of the 21st ACM Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 216–228. ISBN: 9781450391290. URL: https://doi.org/10.1145/3487552.3487847.

[41] Yongzhou Chen, Ruihao Yao, Haitham Hassanieh, and Radhika Mittal. “Channel-Aware 5G RAN Slicing with Customizable Schedulers”. In: *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. Boston, MA: USENIX Association, Apr. 2023, pp. 1767–1782. ISBN: 978-1-939133-33-5. URL: https://www.usenix.org/conference/nsdi23/presentation/chen-yongzhou.

[42] Yongzhou Chen, Ruihao Yao, Haitham Hassanieh, and Radhika Mittal. “Channel-Aware 5G RAN Slicing with Customizable Schedulers”. In: *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 2023.

[43] Yulong Chen, Junchen Guo, Yimiao Sun, Haipeng Yao, Yunhao Liu, and Yuan He. “ELASE: Enabling Real-time Elastic Sensing Resource Scheduling in 5G vRAN”. In: *IEEE/ACM International Symposium on Quality of Service (IWQoS)*. 2024.

[44] *Coco Dataset*. 2024. URL: https://cocodataset.org/#home.

[45] Nokia Communications. *5G Carrier Aggregation explained*. https://www.nokia.com/about-us/newsroom/articles/5g-carrier-aggregation-explained/#:~:text=Carrier%20Aggregation%20is%20a%20software,enhance%20the%20end%20user%20experience. Accessed: 2023-06-09. 2023.

[46] Android Developers community. *Android Debug Bridge (ADB)*. 2023. URL: https://developer.android.com/studio/command-line/adb.

[47] *CSGO*. 2022. URL: https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/.

[48] *Dash-Industry-Forum, dash.js*. 2021. URL: https://github.com/Dash-Industry-Forum/dash.js.

[49] Haotian Deng, Qianru Li, Jingqi Huang, and Chunyi Peng. “iCellSpeed: increasing cellular data speed with device-assisted cell selection”. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–13.

[50] Haotian Deng, Chunyi Peng, Ans Fida, Jiayi Meng, and Y. Charlie Hu. “Mobility Support in Cellular Networks: A Measurement Study on Its Configurations and Implications”. In: *Proceedings of the Internet Measurement Conference 2018*. 2018, pp. 147–160.

[51] Android Developers. *Media3 Exoplayer RTSP*. 2023. URL: https://developer.android.com/media/media3/exoplayer/rtsp.

[52] Ming Ding, David López Pérez, Athanasios V. Vasilakos, and Wen Chen. “Dynamic TDD transmissions in homogeneous small cell networks”. In: *2014 IEEE International Conference on Communications Workshops (ICC)*. 2014.

[53] Phuc Dinh, Moinak Ghoshal, Dimitrios Koutsonikolas, and Joerg Widmer. “Demystifying Resource Allocation Policies in Operational 5G mmWave Networks”. In: *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2022, pp. 1–10. DOI: 10.1109/WoWMoM54355.2022.00016.

[54] Data Center Dynamics. *Syniverse blames US carrier roaming outage on "signaling storm"*. 2025. URL: https://www.datacenterdynamics.com/en/news/syniverse-blames-us-carrier-roaming-outage-on-signaling-storm.

[55] Ibrahim Elbatal, Umar Danjuma Maiwada, Kamaluddeen Usman Danyaro, and Aliza Bt Sarlan. “Dynamic handover optimization in 5G heterogeneous networks”. In: *Journal of Radiation Research and Applied Sciences* (2025). DOI: 10.1016/j.jrras.2025.101411.

[56] Hisham Elshaer, Federico Boccardi, Mischa Dohler, and Ralf Irmer. “Downlink and uplink decoupling: A disruptive architectural design for 5G networks”. In: *2014 IEEE global communications conference (GLOBECOM)*. IEEE. 2014.

[57] Ericsson. *Building sustainable networks*. 2021. URL: https://www.ericsson.com/4ad58a/assets/local/reports-papers/mobility-report/documents/2021/building-sustainable-networks.pdf.

[58] *EU and China lagging behind in mmWave spectrum*. 2024. URL: https://5gobservatory.eu/eu-and-china-lagging-behind-in-mmwave-spectrum/.

[59] Zeinab Ezzeddine, Ayman Khalil, Besma Zeddini, and Habiba Hafdallah Ouslimani. “A Survey on Green Enablers: A Study on the Energy Efficiency of AI-Based 5G Networks”. In: *Sensors* (2024). DOI: 10.3390/s24144609.

[60] *FDD LTE frequency bands*. 2024. URL: https://www.4g-lte.net/about/lte-frequency-bands/fdd/..

[61] Rostand A. K. Fezeu, Jason Carpenter, Claudio Fiandrino, Eman Ramadan, Wei Ye, Joerg Widmer, Feng Qian, and Zhi-Li Zhang. *Mid-Band 5G: A Measurement Study in Europe and US*. 2023. arXiv: 2310.11000.

[62] *ffmpeg Streaming Documentation*. 2023. URL: http://trac.ffmpeg.org/wiki/StreamingGuide.

[63] Piotr Gawłowicz and Anatolij Zubow. "ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research". In: *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. 2019.

[64] Huaining Ge, Xiangming Wen, Wei Zheng, Zhaoming Lu, and Bo Wang. "A History-Based Handover Prediction for LTE Systems". In: *2009 International Symposium on Computer Network and Multimedia Technology*. 2009, pp. 1–4. DOI: 10.1109/CNMT.2009.5374706.

[65] Lucas Chavarria Gimenez, Maria Carmela Cascino, Maria Stefan, Klaus I. Pedersen, and Andrea F. Cattoni. "Mobility Performance in Slow- and High-Speed LTE Real Scenarios". In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. 2016. DOI: 10.1109/VTCSpring.2016.7504347.

[66] Ionel Gog, Sukrit Kalra, Peter Schafhalter, Matthew A Wright, Joseph E Gonzalez, and Ion Stoica. "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021.

[67] Developed with Google. *WifiRttLocator App*. 2023. URL: https://play.google.com/store/apps/details?id=com.google.android.apps.location.rtt.wifirttlocator.

[68] Agrim Gupta, Adel Heidari, Avyakta Kalipattapu, Ish Kumar Jain, and Dinesh Bharadia. "3 W's of smartphone power consumption: Who, Where and How much is draining my battery?" In: *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. ACM MobiCom '24. Washington D.C., DC, USA: Association for Computing Machinery, 2024, pp. 2248–2250. ISBN: 9798400704895. DOI: 10.1145/3636534.3695905.

[69] Ronny Hadani and Anton Monk. *OTFS: A New Generation of Modulation Addressing the Challenges of 5G*. 2018. arXiv: 1802.02623 [cs.IT]. URL: https://arxiv.org/abs/1802.02623.

[70] Bo Han, Yu Liu, and Feng Qian. "ViVo: visibility-aware mobile volumetric video streaming". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, 2020. DOI: 10.1145/3372224.3380888.

[71] Ahmad Hassan, Shivang Aggarwal, Mohamed Ibrahim, Puneet Sharma, and Feng Qian. "Wixor: Dynamic TDD Policy Adaptation for 5G/xG Networks". In: *Proc. ACM Netw.* 2.CoNEXT4 (Nov. 2024). DOI: 10.1145/3696395.

[72] Ahmad Hassan, Arvind Narayanan, Anlan Zhang, Wei Ye, Ruiyang Zhu, Shuowei Jin, Jason Carpenter, Z. Morley Mao, Feng Qian, and Zhi-Li Zhang. "Vivisecting Mobility Management in 5G Cellular Networks". In: *Proceedings of the ACM SIGCOMM 2022 Conference*. New York, NY, USA: Association for Computing Machinery, 2022. ISBN: 9781450394208. DOI: 10.1145/3544216.3544217.

[73] Ahmad Hassan, Arvind Narayanan, Anlan Zhang, Wei Ye, Ruiyang Zhu, Shuowei Jin, Jason Carpenter, Z. Morley Mao, Feng Qian, and Zhi-Li Zhang. "Vivisecting mobility management in 5G cellular networks". In: *SIGCOMM '22*. 2022. DOI: 10.1145/3544216.3544217.

[74] Ahmad Hassan, Wei Ye, Anlan Zhang, Rostand A. K. Fezeu, Jason Carpenter, Ruiyang Zhu, Shuowei Jin, Myungjin Lee, Akshay Jajoo, Morley Mao, Zhi-Li Zhang, and Feng Qian. "OPCM: Opportunistic Performance-driven Connectivity Management for 5G/xG Networks ". In: *Proc. ACM Netw.* 3.CoNEXT4 (2025). DOI: 10.1145/3768970.

[75] Ananya Hazarika and Mehdi Rahmati. "Towards an Evolved Immersive Experience: Exploring 5G- and Beyond-Enabled Ultra-Low-Latency Communications for Augmented and Virtual Reality". In: *Sensors* 23.7 (2023). DOI: 10.3390/s23073682.

[76] *High-Quality 5G Networks Bring the World Faster to the 5.5G Era*. 2024. URL: https://www.huawei.com/en/news/2024/2/5g-high-quality-network-5g-a#:~:text=Multi%2Dcarrier%20networks%20are%20becoming,all%20now%20multi%2Dcarrier%20capable..

[77] *Hitman2*. 2018. URL: https://store.steampowered.com/app/863550/HITMAN_2/.

[78] Hongji Huang, Song Guo, Guan Gui, Zhen Yang, Jianhua Zhang, Hikmet Sari, and Fumiyuki Adachi. *Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions*. 2019. arXiv: 1904.09673 [eess.SP]. URL: https://arxiv.org/abs/1904.09673.

[79] Adrian Ichimescu, Nirvana Popescu, Eduard C. Popovici, and Antonela Toma. "Energy Efficiency for 5G and Beyond 5G: Potential, Limitations, and Future Directions". In: *Sensors* 24 (2024). DOI: 10.3390/s24227402.

[80] *iperf3 – iperf3 3.9 documentation*. 2021. URL: https://software.es.net/iperf/.

[81] Rostand A K. Fezeu, Claudio Fiandrino, Eman Ramadan, Jason Carpenter, Lilian Coelho de Freitas, Faaiq Bilal, Wei Ye, Joerg Widmer, Feng Qian, and Zhi-Li Zhang. "Unveiling the 5G Mid-Band Landscape: From Network Deployment to Performance and Application QoE". In: *Proceedings of the ACM SIGCOMM 2024 Conference*. 2024, pp. 358–372.

[82] Gaganpreet Kaur, Raman Kumar Goyal, and Rajesh Mehta. “An efficient handover mechanism for 5G networks using hybridization of LSTM and SVM”. In: *Multimedia Tools Appl.* (2022). DOI: 10.1007/s11042-021-11510-x.

[83] Jaehong Kim, Yunheon Lee, Hwijoon Lim, Youngmok Jung, Song Min Kim, and Dongsu Han. “OutRAN: co-optimizing for flow completion time in radio access network”. In: *Proceedings of the 18th International Conference on Emerging Networking EXperiments and Technologies*. CoNEXT ’22. 2022.

[84] Tae-Hyong Kim, Qiping Yang, Jae-Hyoung Lee, Soon-Gi Park, and Yeon-Seung Shin. “A Mobility Management Technique with Simple Handover Prediction for 3G LTE Systems”. In: *2007 IEEE 66th Vehicular Technology Conference*. 2007, pp. 259–263. DOI: 10.1109/VETECF.2007.68.

[85] Volodymyr Kuleshov and Doina Precup. *Algorithms for multi-armed bandit problems*. 2014. arXiv: 1402.6028 [cs.AI]. URL: https://arxiv.org/abs/1402.6028.

[86] Lopamudra Kundu, Xingqin Lin, and Rajesh Gadiyar. “Toward Energy Efficient RAN: From Industry Standards to Trending Practice”. In: *Wireless Commun.* (Feb. 2025). ISSN: 1536-1284. DOI: 10.1109/MWC.010.2400061.

[87] Li Li, Ke Xu, Tong Li, Kai Zheng, Chunyi Peng, Dan Wang, Xiangxiang Wang, Meng Shen, and Rashid Mijumbi. “A measurement study on multi-path TCP with multiple cellular carriers on high speed rails”. In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 2018, pp. 161–175.

[88] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. “Towards Streaming Perception”. In: *Computer Vision – ECCV 2020*. 2020.

[89] Qianru Li and Chunyi Peng. “Reconfiguring Cell Selection in 4G/5G Networks”. In: *2021 IEEE 29th International Conference on Network Protocols (ICNP)*. IEEE. 2021, pp. 1–11.

[90] Qianru Li, Zhehui Zhang, Yanbing Liu, Zhaowei Tan, Chunyi Peng, and Songwu Lu. “CA++: Enhancing Carrier Aggregation Beyond 5G”. In: *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: Association for Computing Machinery, 2023. ISBN: 9781450399906.

[91] Yuanjie Li, Haotian Deng, Jiayao Li, Chunyi Peng, and Songwu Lu. “Instability in Distributed Mobility Management: Revisiting Configuration Management in 3G/4G Mobile Networks”. In: *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. 2016, pp. 261–272.

[92] Yuanjie Li, Haotian Deng, Chunyi Peng, Zengwen Yuan, Guan-Hua Tu, Jiayao Li, and Songwu Lu. "iCellular: Device-Customized Cellular Network Access on Commodity Smartphones". In: *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. Santa Clara, CA: USENIX Association, Mar. 2016, pp. 643–656. ISBN: 978-1-931971-29-4. URL: https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/li-yuanjie.

[93] Yuanjie Li, Qianru Li, Zhehui Zhang, Ghufran Baig, Lili Qiu, and Songwu Lu. "Beyond 5G: Reliable Extreme Mobility Management". In: *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*. SIGCOMM '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 344–358. ISBN: 9781450379557. DOI: 10.1145/3387514.3405873.

[94] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. "MobileInsight: Extracting and Analyzing Cellular Network Information on Smartphones". In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 2016, pp. 202–215.

[95] Yuanjie Li, Jiaqi Xu, Chunyi Peng, and Songwu Lu. "A first look at unstable mobility management in cellular networks". In: *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. 2016, pp. 15–20.

[96] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C Begen, and David Oran. "Probe and adapt: Rate adaptation for HTTP video streaming at scale". In: *IEEE journal on selected areas in communications* 32.4 (2014), pp. 719–733.

[97] Shih-Chieh Lin, Yunqi Zhang, Chang-Hong Hsu, Matt Skach, Md E Haque, Lingjia Tang, and Jason Mars. "The architectural implications of autonomous driving: Constraints and acceleration". In: *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*. 2018, pp. 751–766.

[98] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[99] *linux socket statistics*. 2022. URL: https://man7.org/linux/man-pages/man8/ss.8.html.

[100] Peng Liu, Bozhao Qi, and Suman Banerjee. "Edgeeye: An edge service framework for real-time intelligent video analytics". In: *Proceedings of the 1st international workshop on edge systems, analytics and networking*. 2018, pp. 1–6.

[101] Yanbing Liu, Junpeng Guo, and Chunyi Peng. “Demystifying Secondary Radio Access Failures in 5G”. In: *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*. HOTMOBILE ’24. San Diego, CA, USA: Association for Computing Machinery, 2024, pp. 114–120. ISBN: 9798400704970. DOI: 10.1145/3638550.3641125.

[102] Yanbing Liu and Chunyi Peng. “A Close Look at 5G in the Wild: Unrealized Potentials and Implications”. In: *IEEE International Conference on Computer Communications (INFOCOM’23)*. 2023.

[103] Yanbing Liu and Chunyi Peng. “ Handling Failures in Secondary Radio Access Failure Handling in Operational 5G Networks ”. In: *IEEE Transactions on Mobile Computing* 01 (Oct. 5555), pp. 1–14. ISSN: 1558-0660. DOI: 10.1109/TMC.2024.3477462.

[104] videosdk live. *WebRTC based video conferencing SDK for Android.* 2025. URL: https://github.com/videosdk-live/videosdk-rtc-android-kotlin-sdk-example.

[105] *LiveVideoBroadcaster.* 2024. URL: https://github.com/ant-media/LiveVideoBroadcaster.

[106] M. Carmen Lucas-Estañ and J. Gozalvez. “Sensing-Based Grant-Free Scheduling for Ultra Reliable Low Latency and Deterministic Beyond 5G Networks”. In: *IEEE Transactions on Vehicular Technology* (2022).

[107] Anna Łukowa and Venkatkumar Venkatasubramanian. “Centralized UL/DL Resource Allocation for Flexible TDD Systems With Interference Cancellation”. In: *IEEE Transactions on Vehicular Technology* 68.3 (2019), pp. 2443–2458. DOI: 10.1109/TVT.2019.2893061.

[108] Jiamei Lv, Yi Gao, Zhi Ding, Yuxiang Lin, Xinyun You, Guang Yang, and Wei Dong. “Providing UE-level QoS Support by Joint Scheduling and Orchestration for 5G vRAN”. In: *IEEE International Conference on Computer Communications (INFOCOM)*. 2024.

[109] Kyle MacMillan, Tarun Mangla, James Saxon, and Nick Feamster. “Measuring the Performance and Network Utilization of Popular Video Conferencing Applications”. In: *Proceedings of the 21st ACM Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2021. ISBN: 9781450391290. URL: https://doi.org/10.1145/3487552.3487842.

[110] MediaTek Inc. *5G NR Power Saving Enhancements in Release 17*. Tech. rep. MediaTek, 2022. URL: https://mediatek-marketing.files.svdcdn.com/production/documents/MediaTek_White-Paper-R17-5G.pdf.

[111] MediaTek Inc. *Evaluations and Observations for Release 17 UE Power Saving Enhancements*. Tech. rep. 2023. URL: https://newsletter.mediatek.com/hubfs/R1-2003667_Evaluations%20and%20observations%20for%20R17%20UE%20power%20saving%20enhancements_final.pdf.

[112] Lifan Mei, Jinrui Gou, Yujin Cai, Houwei Cao, and Yong Liu. “Realtime Mobile Bandwidth and Handoff Predictions in 4G/5G Networks”. In: *CoRR* abs/2104.12959 (2021). arXiv: 2104.12959. URL: https://arxiv.org/abs/2104.12959.

[113] Zili Meng, Yaning Guo, Chen Sun, Bo Wang, Justine Sherry, Hongqiang Harry Liu, and Mingwei Xu. “Achieving consistent low latency for wireless real-time communications with the shortest control loop”. In: *Proceedings of the ACM SIGCOMM 2022 Conference*. SIGCOMM ’22. 2022.

[114] *Monsoon High Voltage Power Monitor*. URL: https://www.msoon.com/high-voltage-power-monitor (visited on 08/18/2025).

[115] *Monsoon power monitor*. https://www.msoon.com/LabEquipment/PowerMonitor/. 2022.

[116] Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. “A first look at commercial 5G performance on smartphones”. In: *Proceedings of The Web Conference 2020*. 2020, pp. 894–905.

[117] Arvind Narayanan, Eman Ramadan, Rishabh Mehta, Xinyue Hu, Qingxu Liu, Rostand AK Fezeu, Udhaya Kumar Dayalan, Saurabh Verma, Peiqi Ji, Tao Li, et al. “Lumos5G: Mapping and Predicting Commercial mmWave 5G Throughput”. In: *Proceedings of the ACM Internet Measurement Conference*. 2020, pp. 176–193.

[118] Arvind Narayanan, Eman Ramadan, Rishabh Mehta, Xinyue Hu, Qingxu Liu, Rostand AK Fezeu, Udhaya Kumar Dayalan, Saurabh Verma, Peiqi Ji, Tao Li, et al. “Lumos5G: Mapping and predicting commercial mmWave 5G throughput”. In: *Proceedings of the ACM Internet Measurement Conference*. 2020, pp. 176–193.

[119] Arvind Narayanan, Eman Ramadan, Jacob Quant, Peiqi Ji, Feng Qian, and Zhi-Li Zhang. “5G Tracker: A Crowdsourced Platform to Enable Research Using Commercial 5g Services”. In: *Proceedings of the SIGCOMM ’20 Poster and Demo Sessions*. SIGCOMM ’20. Virtual event: Association for Computing Machinery, 2020, pp. 65–67. ISBN: 9781450380485. DOI: 10.1145/3405837.3411394.

[120] Arvind Narayanan, Eman Ramadan, Jacob Quant, Peiqi Ji, Feng Qian, and Zhi-Li Zhang. “5G tracker: a crowdsourced platform to enable research using commercial 5g services”. In: *Proceedings of the SIGCOMM’20 Poster and Demo Sessions*. 2020, pp. 65–67.

[121] Arvind Narayanan, Muhammad Iqbal Rochman, Ahmad Hassan, Bariq S. Firmansyah, Vanlin Sathya, Monisha Ghosh, Feng Qian, and Zhi-Li Zhang. “A Comparative Measurement Study of Commercial 5G mmWave Deployments”. In: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. 2022, pp. 800–809. DOI: 10.1109/INFOCOM48880.2022.9796693.

[122] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, et al. "A Variegated Look at 5G in the Wild: Performance, Power, and QoE Implications". In: *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 2021, pp. 610–625.

[123] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. "Mahimahi: Accurate Record-and-Replay for HTTP". In: *2015 USENIX Annual Technical Conference (USENIX ATC 15)*. Santa Clara, CA: USENIX Association, July 2015, pp. 417–429. ISBN: 978-1-931971-225. URL: https://www.usenix.org/conference/atc15/technical-session/presentation/netravali.

[124] Nidhi, Albena Mihovska, and Ramjee Prasad. "Overview of 5G New Radio and Carrier Aggregation: 5G and Beyond Networks". In: *2020 23rd International Symposium on Wireless Personal Multimedia Communications (WPMC)*. 2020.

[125] *NR and NG-RAN Overall description*. 2021. URL: https://www.etsi.org/deliver/etsi_ts/138300_138399/138300/16.04.00_60/ts_138300v160400p.pdf.

[126] *NS-3 LTE Module*. 2024. URL: https://www.nsnam.org/docs/models/html/lte.html.

[127] *O-RAN Architecture*. 2023. URL: https://docs.o-ran-sc.org/en/latest/architecture/architecture.html.

[128] *Open Source 5G Implementation*. 2024. URL: https://open5gs.org/..

[129] *Open Source RAN*. 2024. URL: https://github.com/srsRAN.

[130] Metin Ozturk, Mandar Gogate, Oluwakayode Onireti, Ahsan Adeel, Amir Hussain, and Muhammad A. Imran. "A novel deep learning driven, low-cost mobility prediction approach for 5G cellular networks: The case of the Control/Data Separation Architecture (CDSA)". In: *Neurocomputing* 358 (2019), pp. 479–489. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.01.031.

[131] Ioannis Panitsas, Akrit Mudvari, Ali Maatouk, and Leandros Tassiulas. *Predictive Handover Strategy in 6G and Beyond: A Deep and Transfer Learning Approach*. 2024. URL: https://arxiv.org/abs/2404.08113.

[132] Jaeeun Park, Joohyung Lee, Daejin Kim, and Jun Kyun Choi. "Deep Reinforcement Learning Driven Joint Dynamic TDD and RRC Connection Management Scheme in Massive IoT Networks". In: *IEEE Access* (2024).

[133] Natale Patriciello, Sandra Lagen, Biljana Bojovic, and Lorenza Giupponi. "An E2E simulator for 5G NR networks". In: *Simulation Modelling Practice and Theory* 96 (2019), p. 101933.

[134] Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. "PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth". In: *Proceedings 17th International Conference on Data Engineering*. 2001, pp. 215–224. DOI: 10.1109/ICDE.2001.914830.

[135] Chunyi Peng and Yuanjie Li. "Demystify Undesired Handoff in Cellular Networks". In: *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. 2016, pp. 1–9. DOI: 10.1109/ICCCN.2016.7568506.

[136] *Procedures for the 5G System (5GS)*. 2024. URL: https://www.etsi.org/deliver/etsi_ts/123500_123599/123502/15.05.01_60/ts_123502v150501p.pdf.

[137] *Qualcomm QXDM Professional(TM) Tool Quick Start*. 2022. URL: https://www.qualcomm.com/media/documents/files/qxdm-professional-qualcomm-extensible-diagnostic-monitor.pdf.

[138] Qualcomm Technologies, Inc. *5G-Advanced: Release 19 and Beyond*. 2024. URL: https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/5G-A-Rel-19-Presentation.pdf (visited on 08/18/2025).

[139] Tech Radar. *T-Mobile went down – everything we know about this network outage*. 2025. URL: https://www.techradar.com/news/live/tmobile-november-outage.

[140] The Register. *Failure to follow proper procedures caused US-wide AT&T outage, FCC says*. 2025. URL: https://www.theregister.com/2024/07/23/atandt_outage_fcc_report/.

[141] Muhammad Iqbal Rochman, Wei Ye, Zhi-Li Zhang, and Monisha Ghosh. "A Comprehensive Real-World Evaluation of 5G Improvements over 4G in Low-and Mid-Bands". In: *2024 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE. 2024, pp. 257–266.

[142] Samsung. *4G-5G Interworking*. 2017. URL: https://images.samsung.com/is/content/samsung/p5/global/business/networks/insights/white-paper/4g-5g-interworking/global-networks-insight-4g-5g-interworking-0.pdf.

[143] Samsung. *5G Standalone Architecture*. 2021. URL: https://images.samsung.com/is/content/samsung/assets/global/business/networks/insights/white-papers/0107_5g-standalone-architecture/5G_SA_Architecture_Technical_White_Paper_Public.pdf.

[144] Samsung Electronics. *Energy-Saving for 6G Network: From Always-ON to Smart-ON*. 2025. URL: https://research.samsung.com/blog/Energy-Saving-for-6G-Network-Part-II-From-Always-ON-to-Smart-ON.

[145] Samsung Electronics. *Exynos 2400 Mobile Processor*. URL: https://semiconductor.samsung.com/processor/mobile-processor/exynos-2400/ (visited on 08/18/2025).

[146] Samsung Semiconductor Global. *Exynos Mobile Processor*. 2025. URL: https://semiconductor.samsung.com/processor/mobile-processor/ (visited on 10/01/2025).

[147] William Sentosa, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Haitham Hassanieh, and Bruce Maggs. "DChannel: Accelerating Mobile Applications With Parallel High-bandwidth and Low-latency Channels". In: *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 2023.

[148] *Serving Models*. 2024. URL: https://www.tensorflow.org/tfx/guide/serving.

[149] Chandan Kumar Sheemar, Leonardo Badia, and Stefano Tomasin. "Game-Theoretic Mode Scheduling for Dynamic TDD in 5G Systems". In: *IEEE Communications Letters* 25.7 (2021), pp. 2425–2429. DOI: 10.1109/LCOMM.2021.3073908.

[150] Chenhua Shen. "Analysis of detrended time-lagged cross-correlation between two nonstationary time series". In: *Physics Letters A* 379.7 (2015), pp. 680–687.

[151] *Simple RTSP Server*. 2023. URL: https://github.com/aler9/rtsp-simple-server.

[152] *Snapdragon 865 5G Mobile Platform*. 2022. URL: https://www.qualcomm.com/products/snapdragon-865-5g-mobile-platform.

[153] *Snapdragon 888 5G Mobile Platform*. 2022. URL: https://www.qualcomm.com/products/snapdragon-888-5g-mobile-platform.

[154] Jing Song, Qingyang Song, Ya Kang, Lei Guo, and Abbas Jamalipour. "QoE-Driven Distributed Resource Optimization for Mixed Reality in Dynamic TDD Systems". In: *IEEE Transactions on Communications* 70.11 (2022), pp. 7294–7306. DOI: 10.1109/TCOMM.2022.3208113.

[155] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K Sitaraman. "BOLA: Near-optimal bitrate adaptation for online videos". In: *IEEE/ACM transactions on networking* 28.4 (2020), pp. 1698–1711.

[156] srslte. *srsRan 4G*. 2024. URL: https://github.com/srsran/srsRAN_4G.

[157] *srsRAN 4G with ZMQ Virtual Radios*. 2024. URL: https://docs.srsran.com/projects/4g/en/latest/app_notes/source/zeromq/source/index.html.

[158] *srsRAN: A customisable solution for Private Enterprise 5G*. 2024. URL: https://srs.io/srsran-enterprise-5g/.

[159] *Steam Link*. 2022. URL: https://store.steampowered.com/app/353380/Steam_Link/.

[160] *Steam remote play*. 2022. URL: https://partner.steamgames.com/doc/features/remoteplay.

[161] Hongguang Sun, Matthias Wildemeersch, Min Sheng, and Tony Q. S. Quek. “D2D Enhanced Heterogeneous Cellular Networks With Dynamic TDD”. In: *IEEE Transactions on Wireless Communications* (2015).

[162] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[163] Zhaowei Tan, Jinghao Zhao, Yuanjie Li, Yifei Xu, and Songwu Lu. “Device-Based LTE Latency Reduction at the Application Layer”. In: *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. 2021.

[164] Fengxiao Tang, Yibo Zhou, and Nei Kato. “Deep Reinforcement Learning for Dynamic Uplink/Downlink Resource Allocation in High Mobility 5G HetNet”. In: *IEEE Journal on Selected Areas in Communications* (2020).

[165] *TCP BBR*. 2022. URL: https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control.

[166] *TCP Cubic*. 2022. URL: https://datatracker.ietf.org/doc/html/rfc8312.

[167] FoneArena News Team. *Qualcomm and Ericsson showcase AI-powered wireless and 6G advances at MWC 2025*. https://www.fonearena.com/blog/447227/qualcomm-ai-powered-wireless-6g-advances-mwc-2025.html. Accessed: 2025-05-31. 2025.

[168] Michael ThelanderMichael Thelander. *This is why I TURNED OFF 5G*. https://www.linkedin.com/posts/michaelthelander_this-is-why-i-turned-off-5gfor-someone-activity-7115871320852103168-I7_L/. Accessed: 2024-09-05. 2023.

[169] Van Dat Tuong, Nhu-Ngoc Dao, Wonjong Noh, and Sungrae Cho. “Deep Reinforcement Learning-Based Hierarchical Time Division Duplexing Control for Dense Wireless and Mobile Networks”. In: *IEEE Transactions on Wireless Communications* (2021).

[170] *Understanding RTMP (Real-Time Messaging Protocol) for Seamless Streaming*. 2024. URL: https://medium.com/@usamawizard/understanding-rtmp-real-time-messaging-protocol-for-seamless-streaming-7d7d963ba0ef.

[171] USRP. *USRP B210 Universal Software Radio Peripheral*. 2024. URL: https://www.ettus.com/all-products/ub210-kit/.

[172] Deepak Vasisht, Swarun Kumar, Hariharan Rahul, and Dina Katabi. “Eliminating Channel Feedback in Next-Generation Cellular Networks”. In: *Proceedings of the 2016 ACM SIGCOMM Conference*. SIGCOMM ’16. Florianopolis, Brazil: Association for Computing Machinery, 2016, pp. 398–411. ISBN: 9781450341936. DOI: 10.1145/2934872.2934895.

[173] Andressa Vergutz, Guevara Noubir, and Michele Nogueira. “Reliability for smart healthcare: A network slicing perspective”. In: *IEEE Network* 34.4 (2020), pp. 91–97.

[174] Verizon. *Verizon mmWave Coverage*. 2024. URL: https://www.verizon.com/coverage-map/.

[175] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 7464–7475.

[176] Donglin Wang, Anjie Qiu, Sanket Partani, Qiuheng Zhou, and Hans D. Schotten. “Mitigating Unnecessary Handovers in Ultra-Dense Networks through Machine Learning-based Mobility Prediction”. In: *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. 2023. DOI: 10.1109/VTC2023-Spring57618.2023.10200542.

[177] Jing Wang, Yufan Zheng, Yunzhe Ni, Chenren Xu, Feng Qian, Wangyang Li, Wantong Jiang, Yihua Cheng, Zhuo Cheng, Yuanjie Li, et al. “An Active-Passive Measurement Study of TCP Performance over LTE on High-speed Rails”. In: *The 25th Annual International Conference on Mobile Computing and Networking*. 2019, pp. 1–16.

[178] *XCAL: PC based Advanced 5G Network Optimization Solution*. 2024. URL: https://www.accuver.com/products/network-optimization/XCAL.

[179] Yaxiong Xie. *NG-scope*. 2024. URL: https://github.com/YaxiongXiePrinceton/NG-Scope.

[180] Yaxiong Xie and Kyle Jamieson. “Ng-scope: Fine-grained telemetry for nextg cellular networks”. In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6.1 (2022), pp. 1–26.

[181] Dongzhu Xu, Anfu Zhou, Guixian Wang, Huanhuan Zhang, Xiangyu Li, Jialiang Pei, and Huadong Ma. “Tutti: coupling 5G RAN and mobile edge computing for latency-critical video analytics”. In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. MobiCom ’22. 2022.

[182] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. “Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption”. In: *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 2020, pp. 479–494.

[183] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. "Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption". In: *SIGCOMM '20*. 2020. DOI: 10.1145/3387514.3405882.

[184] Shichang Xu, Ashkan Nikravesh, and Z Morley Mao. "Leveraging context-triggered measurements to characterize lte handover performance". In: *International Conference on Passive and Active Network Measurement*. Springer. 2019, pp. 3–17.

[185] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. *SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines*. 2020. arXiv: 1911.06188 [cs.CV].

[186] Vijaya Yajnanarayana, Henrik Ryden, and Laszlo Hevizi. "5G Handover using Reinforcement Learning". In: *2020 IEEE 3rd 5G World Forum (5GWF)*. IEEE, 2020. DOI: 10.1109/5gwf49715.2020.9221072.

[187] Mu Yan, Gang Feng, Jianhong Zhou, Yao Sun, and Ying-Chang Liang. "Intelligent Resource Scheduling for 5G Radio Access Network Slicing". In: *IEEE Transactions on Vehicular Technology* (2019).

[188] Xinlei Yang, Hao Lin, Zhenhua Li, Feng Qian, Xingyao Li, Zhiming He, Xudong Wu, Xianlong Wang, Yunhao Liu, Zhi Liao, et al. "Mobile access bandwidth in practice: Measurement, analysis, and implications". In: *Proceedings of the ACM SIGCOMM 2022 Conference*. 2022, pp. 114–128.

[189] Wei Ye, Xinyue Hu, Steven Sleder, Anlan Zhang, Udhaya Kumar Dayalan, Ahmad Hassan, Rostand A. K. Fezeu, Akshay Jajoo, Myungjin Lee, Eman Ramadan, Feng Qian, and Zhi-Li Zhang. "Dissecting Carrier Aggregation in 5G Networks: Measurement, QoE Implications and Prediction". In: *Proceedings of the ACM SIGCOMM 2024 Conference*. ACM SIGCOMM '24. Sydney, NSW, Australia: Association for Computing Machinery, 2024, pp. 340–357. ISBN: 9798400706141. DOI: 10.1145/3651890.3672250.

[190] Bo Yu, Liuqing Yang, Hiroyuki Ishii, and Sayandev Mukherjee. "Dynamic TDD Support in Macrocell-Assisted Small Cell Architecture". In: *IEEE Journal on Selected Areas in Communications* (2015).

[191] Ruozhou Yu, Dejun Yang, and Hao Zhang. "Edge-Assisted Collaborative Perception in Autonomous Driving: A Reflection on Communication Design". In: *2021 IEEE/ACM Symposium on Edge Computing (SEC)*. 2021, pp. 371–375. DOI: 10.1145/3453142.3491413.

[192] Zhehui Zhang, Yanbing Liu, Qianru Li, Zizheng Liu, Chunyi Peng, and Songwu Lu. "Dependent Misconfigurations in 5G/4.5G Radio Resource Control". In: *Proc. ACM Netw.* 1.CoNEXT1 (July 2023). DOI: 10.1145/3595288.

[193] Xiao Zhu, Subhabrata Sen, and Z Morley Mao. “Livelyzer: analyzing the first-Mile ingest performance of live video streaming”. In: *Proceedings of the 12th ACM Multimedia Systems Conference*. 2021.

[194] *Zoom*. 2022. URL: https://zoom.us/.