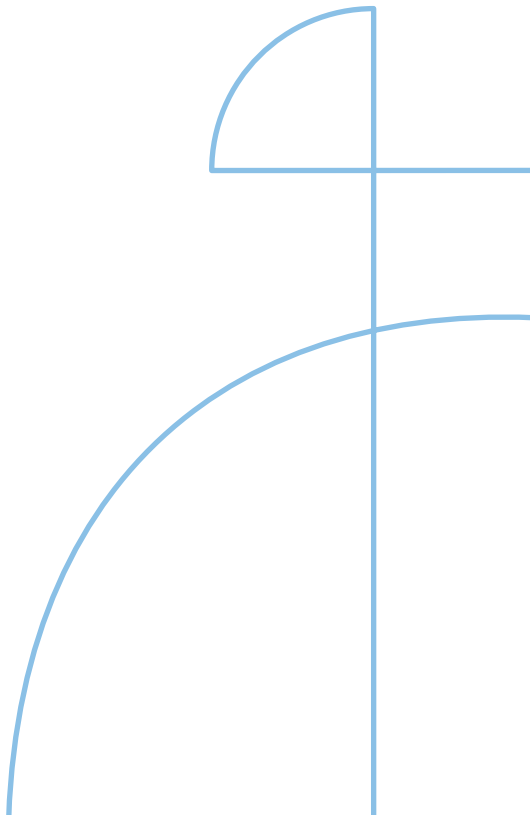
Doctoral Thesis in Electrical Engineering

Predictability, Prediction, and Control of Latency in 5G and Beyond: From Theoretical to Data-Driven Approaches

SAMIE MOSTAFAVI

KTH ROYAL INSTITUTE OF TECHNOLOGY



Predictability, Prediction, and Control of Latency in 5G and Beyond: From Theoretical to Data-Driven Approaches

SAMIE MOSTAFAVI

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Monday the 9th of June 2025, at 10:00 a.m. in D3, Lindstedtvägen 9, Stockholm.

Doctoral Thesis in Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden 2025

TRITA-EECS-AVL-2025:54
ISBN 978-91-8106-285-4

Printed by: Universitetsservice US-AB, Sweden 2025

Abstract

The explosive growth of mobile communication and the proliferation of real-time applications, such as industrial automation and extended reality (XR), have created unprecedented demands for ultra-reliable low-latency communication (URLLC) in wireless networks. For example, in industrial closed-loop control systems, data must be transmitted within a target delay of at most a few milliseconds; violations can lead to costly failures and, therefore, must occur with probabilities below 0.0001 (or, reliability above 0.9999). This dissertation addresses the critical challenge of end-to-end latency prediction and control in these dynamic and stochastic environments, bridging the gap between the inherent randomness of wireless communication and the deterministic performance guarantees required by time-sensitive applications. In this thesis, we adopt a twofold approach, combining rigorous theoretical analysis with practical, data-driven methodologies. First, we introduce a framework for analyzing predictability that quantifies the inherent limits of latency forecasting in communication networks. Through analysis of Markovian systems, including single-hop and multi-hop queues, exact expressions and spectral-based upper bounds for predictability are derived, revealing the crucial influence of network topology, state transitions, and observation defects.

Building on this foundation, we developed and implemented data-driven techniques for probabilistic delay prediction. A key contribution is a tail-optimized prediction method that integrates Extreme Value Theory (EVT) within a mixture density network framework, significantly enhancing the accuracy of predicting rare, high-latency events critical for URLLC. To demonstrate the practical utility of these predictions, "Delta," a novel active queue management scheme, is introduced. Delta integrates real-time delay violation probability predictions into packet-dropping decisions, dynamically adapting to delay variations and significantly reducing delay violations.

To validate these approaches, the ExPECA testbed and EDAF framework were developed, enabling fine-grained delay measurement and decomposition in real 5G systems. Extensive experiments on both commercial off-the-shelf 5G and software-defined radio-based OpenAirInterface platforms confirmed the superior accuracy and efficiency of the proposed EVT-enhanced models. Furthermore, temporal prediction models, leveraging LSTM and Transformer architectures, were developed and shown to achieve higher accuracy compared to the baseline approaches in real 5G network experiments, capturing the time-varying dynamics of wireless networks and providing accurate multi-step forecasts.

This dissertation advances latency prediction and control for wireless networks, offering both theoretical foundations and practical solutions for time-sensitive applications. These findings have significant implications for designing and operating next-generation wireless networks, paving the way for more dependable communication. Future work should focus on integrating these prediction models to optimize the network and extending the framework to encompass broader quality of service metrics and emerging wireless technologies.

Sammanfattning

Den explosionsartade tillväxten av mobil kommunikation och spridningen av realtidsapplikationer, såsom industriell automation och utökad verklighet (XR), har skapat enastående krav på ultratillförlitlig kommunikation med låg fördröjning (URLLC) i trådlösa nätverk. Till exempel måste data i industriella slutna styrsystem överföras inom en deadline på högst några millisekunder; överträdelser kan leda till kostsamma fel och måste därför inträffa med sannolikheter under 0,0001 (eller, en tillförlitlighet över 0,9999). Denna avhandling behandlar den kritiska utmaningen att prediktera och kontrollera fördröjningen mellan sändare till mottagare i dessa dynamiska och stokastiska miljöer, och minskar skillnaden mellan den inneboende slumpmässigheten i trådlös kommunikation och de deterministiska prestandagarantier som krävs av tidskänsliga applikationer. I denna avhandling antas en tvådelad metod som kombinerar noggrann teoretisk analys med praktiska, datadrivna metoder. Först introduceras ett ramverk för att analysera förutsägbarhet som kvantifierar de inneboende gränserna för fördröjningsprognoser i kommunikationsnätverk. Genom att studera Markovsystem, däribland enkel- och multihoppköer, härleds exakta uttryck och spektrumbaserade övre gränser för förutsägbarhet, vilket belyser hur nätverkstopologi, tillståndsövergångar och observationsdefekter påverkar resultaten.

Utifrån denna grund utvecklades och implementerades datadrivna tekniker för probabilistisk fördröjningsprediktion. Ett viktigt bidrag är en metod för prediktion som integrerar extremvärdesteori (EVT) i ett ramverk för blandningstäthetsnätverk, vilket avsevärt förbättrar förmågan att prediktera sällsynta, höga fördröjningar som är avgörande för URLLC. För att demonstrera den praktiska nyttan av dessa prediktioner presenteras ”Delta,” ett nytt aktivt köhanteringssystem. Delta integrerar, i realtid, prediktioner av sannolikheten för fördröjningsöverträdelser i beslutsprocessen för paketborttagning, vilket minskar fördröjningsöverträdelser avsevärt.

För att validera dessa metoder utvecklades testbädden ExPECA och ramverket EDAF, som möjliggör högupplösta mätningar och uppdelning av fördröjningens komponenter i verkliga 5G-system. Omfattande experiment på både kommersiell 5G-utrustning och mjukvarudefinierade radioplattformar baserade på OpenAirInterface bekräftade den förbättrade noggrannheten och effektiviteten hos de föreslagna EVT-förbättrade modellerna. Vidare utvecklades temporala prediktionsmodeller som använder LSTM- och Transformer-arkitekturer som visade högre träffsäkerhet än referensmetoder i verkliga 5G-nätverksexperiment, då de fångar de tidsvarierande dynamikerna i trådlösa nätverk och möjliggör exakta flerstegsprognoser.

Denna avhandling driver framåt forskningen om fördröjningsprediktion och -kontroll i trådlösa nätverk och erbjuder både teoretiska grunder och praktiska lösningar för tidskänsliga applikationer. Resultaten har stor betydelse för utformningen och driften av nästa generations trådlösa nätverk och banar väg för mer pålitlig kommunikation. Framtida arbete ska/borde/kan (will/should/can) fokusera på att integrera dessa prediktionsmodeller för att optimera nätverket, och utvidga ramverket till att omfatta bredare kvalitetsmätningar och nya trådlösa teknologier.

List of Included Publications

The following papers are included in this thesis:

[B] Seyed Samie Mostafavi, György Dán, and James Gross, "Data-driven end-to-end delay violation probability prediction with extreme value mixture models", in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, San Jose, CA, USA: IEEE, Dec. 2021, pp. 416–422.

[F] Samie Mostafavi, Gourav Prateek Sharma, and James Gross, "Data-driven latency probability prediction for wireless networks: Focusing on tail probabilities", in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 4338–4344.

[C] Samie Mostafavi, Neelabhro Roy, György Dán, and James Gross, "Active queue management with data-driven delay violation probability predictors", in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 6371–6376.

[D] Samie Mostafavi, Vishnu Narayanan Moothedath, Stefan Rönngren, Neelabhro Roy, Gourav Prateek Sharma, Sangwon Seo, Manuel Olguin Munoz, and James Gross, "Expeca: An experimental platform for trustworthy edge computing applications", in *Proceedings of the Eighth ACM/IEEE Symposium on Edge Computing*, ser. SEC '23, Wilmington, DE, USA: Association for Computing Machinery, 2024, pp. 294–299.

[E] Samie Mostafavi, Marius Tillner, Gourav Prateek Sharma, and James Gross, "Edaf: An end-to-end delay analytics framework for 5G-and-beyond networks", in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Vancouver, BC, Canada: IEEE, May 2024, pp. 1–6.

[A] Samie Mostafavi, Simon Egger, György Dán, and James Gross, *Predictability of performance in communication networks under markovian dynamics*, under revision, 2024. arXiv: `2408.13196 [cs.NI]`, preprint arXiv:2408.13196.

[G] Samie Mostafavi, Gourav Prateek Sharma, Ahmad Traboulsi, and James Gross, *Probabilistic delay forecasting in 5G using recurrent and attention-based architectures*, under revision, 2025. arXiv: 2503.15297 [cs.NI], preprint arXiv:2503.15297.

In addition to the aforementioned papers, the following papers have also been co-authored by the author of this thesis:

[I] Manuel Olguin Munoz, Seyed Samie Mostafavi, Vishnu N. Moothedath, and James Gross, "Ainur: A framework for repeatable end-to-end wireless edge computing testbed research", in *European Wireless 2022; 27th European Wireless Conference*, Dresden, Germany: VDE, Sep. 2022, pp. 1–7.

[II] Neelabhro Roy, Samie Mostafavi, and James Gross, "Semantically optimized end-to-end learning for positional telemetry in vehicular scenarios", in *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Montreal, QC, Canada: IEEE, Jun. 2023, pp. 425–430.

[III] Panagiotis Nikolaidis, Samie Mostafavi, James Gross, and John Baras, *A proof of concept resource management scheme for augmented reality applications in 5G systems*, Accepted to publish in IEEE International Symposium on Dynamic Spectrum Access Networks, Jan. 2025. arXiv: 2501.01398 [cs.NI], preprint arXiv:2501.01398.

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. James Gross, for his support, encouragement, and patience. His consistent guidance and ambitious ideas at every stage of the research have helped me to complete the PhD journey successfully. He helped me grow as a researcher and as a person in countless ways, and I will always be grateful for that. I want to thank my co-supervisor, Prof. György Dán, for his guidance and thorough reviews of our papers. I am grateful to my co-author, Dr. Gourav Prateek Sharma, for the helpful discussions; his ability to track the related works and hands-on skills supported this research greatly.

I would also like to thank everybody involved in the defence of this thesis. I am very grateful to Prof. Roberto Verdone of the University of Bologna for agreeing to act as my dissertation opponent. My defense will certainly benefit significantly from his expertise and insights. I want to extend my sincere thanks to my grading committee, consisting of Prof. Viktoria Fodor (KTH Royal Institute of Technology), Prof. Holger Rootzén (Chalmers Unversity of Technology), and Dr. Serveh Shalmashi (Ericsson), for investing time and effort into reading and grading this dissertation. Special thanks also go to Prof. Mats Bengtsson (KTH Royal Institute of Technology) for advance-reviewing this work and to Prof. Mikael Skoglund, also of KTH, for agreeing to act as chairperson for the defense.

During my doctoral journey, I was fortunate to work on several projects alongside exceptional research groups that enriched both my knowledge and perspective. I am particularly thankful to Prof. Hongwei Zhang (Iowa State University), Dr. Joachim Sachs (Ericsson), Dr. Bryan Donyanavard (San Diego State University), and Prof. John Baras (University of Maryland) for fruitful collaborations. I would also like to express my sincere appreciation to Marius Tillner, Panagiotis Nikolaidis, Simon Egger, and Arash Sahbafard for the rewarding collaborations we shared.

Moreover, I extend my thanks to my best friend and fellow sailor, Sina Sheikholeslami, who has been my trusted machine-learning expert whenever I struggled with implementation challenges. I'm likewise grateful to my colleagues Martin Lindström, Sara Saeidian, Anhubab Ghosh, Vishnu Narayanan Moothedath and

Amirreza Zamani whose insightful discussions and generous expertise in analytical and theoretical matters helped me along this journey. Special thanks to Martin for the excellent translation of my thesis abstract into Swedish and Sahar for patiently answering all my defense-related questions.

I feel incredibly fortunate to have pursued my PhD with such welcoming, helpful, and engaging colleagues in the Information Science and Engineering (ISE) division of KTH. A heartfelt thank-you goes to the fantastic crew on the 7th floor—who made PhD life fun and memorable for me. You rock, Vishnu, Martin, Sangwon, Neel, Afroditi, Fatemeh, Adarsh, Maryam, Stefan, Gourav, Panos, Jacob, Sahar, and Niloufar. Thanks for all the wonderful memories we created together on this journey. Moreover, I would like to thank Javier, Wendi, Sara, Leo, Steve, Anubhab, Aris, Amaury, Antoine, Borja, Abolfazl, Jeannie, Movitz, Sajad, Shudi, Gustav, Raghav, Yingzhuo, Mengyuan, and Xuechun.

Words cannot describe how grateful I am to Parmiss. Her companionship made this achievement possible. She stood by me during hard times and makes every day we share feel special. I am incredibly thankful to my dear friends Parastu, Albin, Shahab, Romina, Amir, Aida, Oskar, Sara, Sina, Avenia, Shima, Farhad, Mehdi, Saba, and Ali. The parties, trips, and hangouts we've shared have been an invaluable part of my everyday life ever since I moved to Sweden and throughout my PhD. I'm deeply grateful to my long-time friends, Alireza, Masoud, Hossein, Nader, Mohammadreza, Ali, and Sina for their friendship and support, which have continued all these years despite our geographical distance.

Last but not least, my profound and heartfelt thanks go to my family: Maman Nasrin, Baba Javad, Dadash Salman, Dadash Ehsan, and Maryam Jan. Words can't capture how much your love and selfless support have meant to me. Mom and Dad, every call home, every encouraging message, and every quiet sacrifice you made behind the scenes carried me through the tough moments and brought me to this milestone. I owe this as much to you as to my own efforts-thank you from the bottom of my heart.

Seyed Samie Mostafavi
Stockholm, April 2025

List of Acronyms

3GPP	3rd generation partnership project
5GAA	5G automotive association
ACK	acknowledgement
AMF	access and mobility management function
API	application programming interface
AQM	active queue management
ARQ	automatic repeat request
BLER	block error rate
CCDF	complementary cumulative censity function
CDE	conditional density estimation
CDF	cumulative density function
CoDel	controlled delay
COTS	commercial off-the-shelf
CPS	cyber-physical system
CPU	central processing unit
CQI	channel quality indicator
DRL	deep reinforcement learning
DVP	delay violation probability
E2E	end-to-end
EDAF	end-to-end delay analytics framework
EMM	extreme mixture model
EVM	extreme value mixture model
EVT	extreme value theory
ExPECA	experimental platform for edge computing applications

FDD	frequency division duplex
FEC	forward error correction
FIFO	first-in, first-out
FPS	frames per second
GEV	generalized extreme value distribution
GMEVM	Gaussian mixture extreme value model
GMM	Gaussian mixture model
gNB	next generation NodeB
GPD	generalized pareto distribution
GPS	global positioning system
GPU	graphical processing unit
HARQ	hybrid automatic repeat request
HITL	human-in-the-loop
i.i.d.	independent and identically distributed
IoT	internet of things
IP	Internet protocol
KDE	kernel density estimation
KL-divergence	Kullback-Leibler divergence
KPI	key performance indicator
LLM	large language model
LSTM	long short-term memory
LTE	long-term evolution
MAC	medium access control
MAE	mean absolute error
MCS	modulation and coding scheme
MDN	mixture density network
MEC	mobile edge computing
ML	machine learning
MLE	maximum likelihood estimation
MLP	multi-layer perceptron
mMTC	massive machine type communication
mmWave	millimeter-wave
NACK	negative acknowledgement
NLL	negative log-likelihood
NLMT	network latency measurement tool
NWDAF	network data analytics function
OAI	OpenAirInterface
OFDM	orthogonal frequency division multiplexing
OMM	observable markov model
OWD	one-way delay
PDCP	packet data convergence protocol
PDF	probability density function
PDU	protocol data unit
PE	permutation entropy

PIE	proportional integral enhanced
PoT	peaks over thresholds
PRB	physical resource block
PTP	precision time protocol
QoS	quality of service
RAM	random access memory
RAN	radio access network
RED	random early detection
RLC	radio link control
RMSE	root mean square error
RNN	recurrent neural network
RTT	round-trip time
SDR	software-defined radio
SGD	stochastic gradient descent
SINR	signal to interference and noise ratio
SMF	session management function
SN	sequence number
SRS	software radio systems
TBS	transport block size
TDD	time division duplex
TSN	time sensitive networking
TV	total variation
UAV	unmanned aerial vehicle
UDP	user datagram protocol
UE	user equipment
UPF	user plane function
URLLC	ultra-reliable low-latency communication
V2X	vehicle-to-everything
WCA	wearable cognitive assistance
XR	extended reality

Table of Contents

Part I

Summary

CHAPTER **1**

Introduction

In recent decades, the explosive growth of mobile communication has catalyzed a paradigm shift in the design and operation of modern wireless networks. The rapid expansion of the mobile user base, together with the proliferation of real-time applications—ranging from industrial automation to immersive extended reality (XR) experiences—has dramatically intensified the demand for communication services that guarantee exceptionally low delay and near-perfect reliability [1, 2]. Many of these applications operate under strict end-to-end delay deadlines in a closed-loop manner, where sensor data is captured, processed, and acted upon within a very tight time window. This process involves not only the communication delay over the wireless network but also the computation delay incurred by processing tasks on the supporting infrastructure. For instance, in industrial automation, any lag in the control loop can disrupt coordinated processes and jeopardize operational safety. In remote surgery, even a slight delay between a surgeon's command and the robotic response could have severe consequences. Similarly, immersive XR systems demand rapid feedback to ensure a seamless and engaging user experience, making both communication and computation latencies critical performance factors.

The first paradigm shifts in addressing these challenges began in the context of edge computing. Originally conceived to alleviate the latency and bandwidth limitations inherent to centralized cloud architectures, edge computing has evolved into a robust ecosystem of distributed resources that process data much closer to the end-user [2, 3]. By reducing the physical distance over which data must travel and by offloading computation from distant data centers, edge computing not only minimizes transmission delays but also eases network congestion—thereby providing a critical foundation for real-time, closed-loop applications. Building upon this foundation, the next major trend was the emergence of ultra-reliable

low-latency communication (URLLC). As a cornerstone of 5G cellular networks, URLLC builds on the legacy of earlier quality-of-service advancements to achieve delay bounds as tight as 1–10 ms while maintaining reliability levels that typically exceed 99.99% [4]. This stringent performance is indispensable for mission-critical applications, where even minimal delays can lead to catastrophic outcomes. Still, the need for advanced technologies to support these applications more efficiently remains. Market projections further underscore this trend; for example, the edge computing market is forecast to grow from USD 16.45 billion in 2023 to USD 155.90 billion by 2030 [5]. Such forecasts highlight the urgency of developing innovative ways to efficiently support a heterogeneous mix of devices, applications, and end-to-end delay requirements for the next generations of the wireless networks.

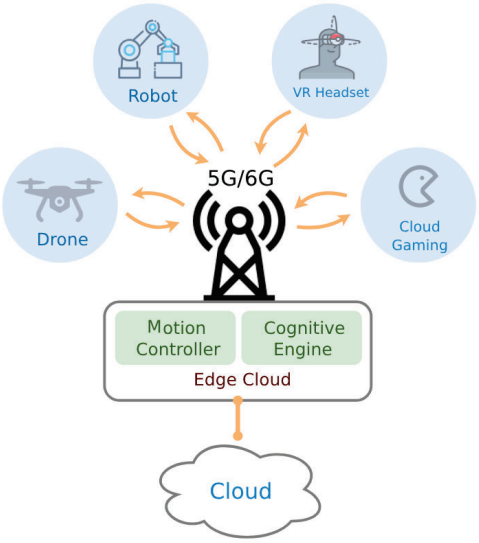


Figure 1.1: The convergence of 5G/6G and edge computing to support emerging real-time, closed loop applications, where the edge cloud, with the motion control and cognitive engine minimize the computation delay, and the 5G/6G guarantee low delay and reliable communication.

To accommodate the diverse needs of these emerging services, the research community has progressively developed tailored quality of service (QoS) criteria. Standardized by 3GPP in the context of 5G and continually refined for the upcoming 6G networks, these criteria define explicit delay targets and reliability thresholds that directly constrain the delay violation probability (DVP) of the system. For example, industrial control applications typically require a delay target of 10 ms and a reliability of 99.99% [6, 7]. Building on these developments, the concept of

predictive QoS has emerged as a critical innovation in this area. Predictive QoS mechanisms empower wireless networks to forecast future performance metrics and deliver advance notifications to network operators and service consumers [8]. By anticipating potential degradations in network conditions, these mechanisms enable preemptive adjustments in resource allocation, thereby mitigating the risk of service interruptions. In the realm of edge computing, the European Telecommunications Standards Institute (ETSI) has introduced dedicated frameworks for predictive QoS support, underscoring the growing importance of anticipatory network management in distributed computing environments [9]. Concurrently, significant research efforts in 6G are focusing on leveraging advanced predictive models to manage dynamic network conditions, particularly in highly mobile or rapidly changing scenarios [10, 11]. According to these initiatives, establishing robust delay prediction mechanisms is imperative—not only to ensure that networks satisfy the strict quality-of-service demands of mission-critical applications but also to avoid the inefficiencies associated with over-provisioning.

Wireless networks inherently pose significant challenges for delay prediction due to their dynamic nature. The stochastic behavior of wireless channels—driven by interference, dynamic fading, and rapid user mobility—injects substantial uncertainty into end-to-end delay measurements [12]. Moreover, the multi-layered architecture of modern wireless systems complicates the prediction task further. Instead of being governed by a single factor, overall delay results from the cumulative impact of various interdependent processes, such as queuing at transmission nodes, scheduling and routing decisions within the network, and retransmission mechanisms, which adds variable delays based on current channel conditions and interference levels. In addition, the highly dynamic nature of network conditions presents another significant obstacle. Wireless environments are characterized by rapidly fluctuating parameters—such as instantaneous signal quality, evolving user mobility patterns, and varying traffic loads—that together create a complex, non-stationary input space. Capturing both the short-term fluctuations and the long-term trends in this high-dimensional space necessitates the development of sophisticated predictive models capable of seamlessly integrating diverse data sources.

Among the notable approaches for delay prediction, model-driven methods, rooted in queuing theory and network calculus, offer a structured framework to capture the statistical properties of end-to-end delay [13]. These analytical models rely on specific assumptions about wireless channel conditions, the independence of multihop links, and stationarity, enabling the derivation of closed-form delay characterizations. While such models can effectively characterize wireless network delay under idealized conditions, their practical applicability is often debated due to the restrictive nature of these assumptions [14]. In the complex, high-dimensional, and non-stationary environments of modern wireless networks, these model-driven approaches may fall short of capturing the full spectrum of delay behaviors. To bridge this gap, data-driven techniques have emerged as a powerful complement. The rapid advancement of machine learning and artificial intelligence has revolutionized the way we approach complex prediction tasks, enabling the extraction of

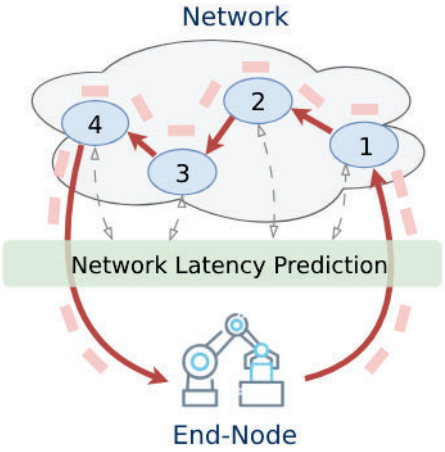


Figure 1.2: Network latency prediction for a time-sensitive closed-loop application.

intricate patterns from vast and unstructured datasets. By harnessing large volumes of historical network data, modern machine learning methods are capable of transforming raw, high-dimensional observations into compact, learnable representations. This data-centric approach not only captures the dynamic dependencies inherent in wireless systems but also adapts more readily to real-world conditions, thereby enhancing the accuracy and robustness of delay predictions.

In this dissertation, we analyze the task of data-driven, machine-learning-based latency prediction for wireless networks, driven by the critical need to bridge the gap between the stringent requirements of real-time applications and the inherent stochasticity of wireless communication. This overarching goal motivates our investigation into several interconnected objectives.

Objective 1. First, we are motivated by the fundamental question of *latency predictability*. Even the most sophisticated prediction models are ultimately limited by the inherent randomness of the wireless medium. Factors such as unpredictable interference, user mobility, and fluctuating channel quality introduce an intrinsic uncertainty into delay outcomes. Understanding and quantifying these limits of predictability is crucial. This motivates our investigation into methods for not only predicting delay but also for assessing the confidence and uncertainty associated with those predictions. This deeper understanding of predictability bounds is essential for setting realistic performance expectations and guiding observability schemes in real-world deployments.

Objective 2. We are motivated by the challenge of *accurate delay prediction itself*. How can we effectively leverage the power of machine learning to forecast end-to-end latency in these complex, dynamic environments? This goes beyond predicting average delays; it necessitates capturing the crucial tail behavior of the

delay distribution, which governs the probability of violating critical performance thresholds. We are driven to explore how different machine learning architectures and techniques can best capture the intricate dependencies within wireless networks, including the impact of factors such as network topology (single-hop vs. multi-hop) and the temporal evolution of network conditions.

Objective 3. The practical and reliable implementation of data-driven delay prediction and control depends on a solid foundation of *data collection and delay measurement*. We aim to develop accurate delay measurement and decomposition methods for 5G networks, along with identifying and collecting relevant information from all network components and processes that impact end-to-end delay. This detailed data is critical for training effective machine learning models and generating insights for delay control in wireless networks.

Objective 4. Ultimately, a core motivation driving this work is the question of *how to leverage these delay predictions to optimize network performance proactively.* Accurate latency forecasts are not merely an end in themselves; they are a powerful tool for enabling intelligent network control and resource management. This motivates us to investigate how predictive insights can be integrated into network data management mechanisms. The goal is to transition from reactive network management to a proactive, prediction-driven approach that can maintain stringent QoS requirements even in the face of inherent wireless variability.

To address these multifaceted challenges, our research methodology follows a two-step approach. We initially focus on abstract queuing-theoretic models to gain a fundamental understanding of the limitations and inherent difficulties in delay prediction and control. This allows us to isolate key factors and derive theoretical insights in a simplified, yet rigorous, setting. Subsequently, we transition to the context of 5G networks, representing one of the most advanced and widely deployed wireless technologies. This real-world setting enables us to further motivate our research, develop practical algorithms, and evaluate their performance under realistic conditions. This progression, from abstract models to a concrete, state-of-the-art wireless system, allows for a comprehensive and robust investigation of latency prediction and management.

1.1 Structure of this Dissertation

This dissertation is divided into two main parts. Part I provides an overview, introducing the research area, reviewing related literature, and summarizing the key contributions. Part II presents the publications that constitute the core of this thesis.

Part I comprises four chapters. Chapter 1 introduces the topic and outlines the dissertation's contributions, setting the context for the research. Chapter 2 reviews relevant prior work and defines the scope of this thesis, establishing a foundation for our contributions. The core of Part I, Chapter 3, details the methods developed and findings obtained during this research, demonstrating their significance to the

field. Finally, Chapter 4 summarizes the key findings and contributions, discusses their implications, and identifies potential avenues for future research.

1.2 Background

1.2.1 Queuing Analysis

The mathematical analysis of queueing systems dates back at least to the year 1909, when A. K. Erlang published his study on "the theory of probabilities and telephone conversations". From the perspective of a telephone company, calls (or *customers* in classical queueing theory) begin at random times and last for a random duration. This randomness can lead to congestion—when all available lines (or servers) are busy—and motivated early analyses to determine the minimum number of lines needed to maintain an acceptable level of service. Similar analyses are now applied to a broad range of applications, from customers waiting at a supermarket checkout to computing tasks queuing for processor time [15, 16].

At its core, a queue is a system where entities (often called *customers* or *tasks*) arrive, wait if necessary, and then receive service from one or more servers. In many applications—such as communication networks where data packets are transmitted between nodes—a task experiences an overall delay that can be decomposed into two main components: the *waiting time* W_n, which is the time a task spends waiting in the queue before service begins, and the *service time* S_n, which is the time taken to complete the service once it has begun. Thus, for task n generated at time T_n, the overall sojourn time (i.e., total delay) is given by

$$Z_n = W_n + S_n. \tag{1.1}$$

In systems where tasks pass through multiple stages (or tandem queues), the end-to-end delay is the sum of the sojourn times across all stages:

$$Z_n = \sum_{i=1}^{N} Z_n^{(i)}, \tag{1.2}$$

with each stage's sojourn time defined similarly as $Z_n^{(i)} = W_n^{(i)} + S_n^{(i)}$. This formulation leads naturally to the study of the delay distribution. For example, the probability that the sojourn time of a task exceeds a target threshold τ is written as

$$\Pr\left(Z_n > \tau\right). \tag{1.3}$$

Analyzing this distribution requires a careful examination of the underlying stochastic processes governing both the arrival of tasks and their service [16].

For the arrival process, in many continuous-time queueing models arrivals are typically assumed to follow a *Poisson process* with an average rate λ. Under this assumption, the interarrival times are independent and identically distributed (i.i.d.)

exponential random variables with probability density function

$$f_A(t) = \lambda e^{-\lambda t}, \quad t \geq 0, \tag{1.4}$$

a fact that follows from the inherent memoryless property of the exponential distribution. In discrete-time models, however, the arrival process is commonly modeled as a *Bernoulli process*, where time is divided into slots and an arrival occurs in each slot with probability λ (and does not occur with probability $1 - \lambda$) [16].

In contrast, the service process describes the time required to complete service for a task once it begins receiving attention from the server. In continuous-time models such as the M/M/1 queue, service times are usually assumed to be exponentially distributed with parameter μ, yielding a probability density function

$$f_S(t) = \mu e^{-\mu t}, \quad t \geq 0. \tag{1.5}$$

The M/M/1 queue is a prototypical model in queueing theory, where the notation follows *Kendall's notation*: the first 'M' denotes that the interarrival times are Markovian (i.e., exponentially distributed), the second 'M' indicates that the service times are also Markovian, and the '1' specifies that there is a single server [17]. The exponential service time is *memoryless*, meaning that the probability of service completion in the next instant is independent of the time already spent in service. In discrete-time models, such as in the Geo/Geo/1 queue, the service process is modeled with a *geometric distribution*. If S denotes the number of time slots required for service, then the probability mass function is given by

$$\Pr(S = k) = \mu(1 - \mu)^{k-1}, \quad k = 1, 2, 3, \ldots, \tag{1.6}$$

where μ represents the probability that the service is completed in any given slot [16].

These assumptions on the arrival and service processes not only simplify the mathematical treatment but also allow for closed-form expressions for key performance metrics. For example, under these assumptions one can derive the probability distribution of the sojourn time Z_n as well as characterize the system behavior—such as through steady-state probabilities or transient state evolution—using tools such as Markov chains [18].

In summary, understanding both the arrival process and the service process is essential in queueing theory, as these processes determine the delay distributions and overall performance of the system. Whether modeled in continuous or discrete time, these fundamental processes underpin the analysis of waiting times, sojourn times, and the resulting performance metrics that are critical for system design and active queue management [16].

Markovian Queues

A significant subset of queueing models assumes that arrival and service processes are Markovian, meaning they possess the memoryless property. In a Markovian system, the future evolution depends only on the current state rather than the

full past history. This property greatly simplifies the analysis. For example, the classical M/M/1 queue is defined by exponential interarrival and service times; the exponential distribution is the only continuous distribution with the memoryless property [17].

In a Markovian queueing system, the evolution of the system is typically modeled as a continuous-time or discrete-time Markov chain. Let X_n denote the state of the system (for example, the number of customers in the queue) at time step n (or $X(t)$ at continuous time t). The transition probability from state i to state j is arranged in a transition probability matrix P where its element on row i and column j corresponds to the the transition probability from state i to state j in one time step as

$$P(i,j) = \Pr(X_{n+1} = j \mid X_n = i). \tag{1.7}$$

In the continuous-time case, this is described via a generator matrix Q [18].

The *transient analysis* of a Markovian system involves computing the state probabilities over a finite time horizon. If the system starts in state i at time step 0, then the probability of being in state j at time step n is given by

$$P^n(i,j) = \Pr(X_n = j \mid X_0 = i), \tag{1.8}$$

which can be obtained by raising the transition matrix P to the nth power (or by solving the corresponding Kolmogorov forward equations in continuous time) [18]. Transient analysis is important because it provides insights into the short-term dynamics and performance of the system, such as evaluating delay distributions or congestion probabilities over finite time intervals [16].

In contrast, the *stationary distribution* (or equilibrium state) is a probability distribution π over the state space that satisfies

$$\pi = \pi P, \tag{1.9}$$

or, in the continuous-time case,

$$\pi Q = 0. \tag{1.10}$$

Under appropriate stability conditions (for example, when the arrival rate is less than the service rate), the system converges to this stationary distribution over time and remains in it thereafter. The stationary distribution is fundamental in queueing theory as it provides long-term performance metrics such as the average number of customers in the system, average waiting times, and the probability of congestion [16, 17].

Studying both the transient and stationary states of Markovian queues is crucial for a comprehensive understanding of system performance. While the stationary state offers insights into the long-run average behavior and stability of the system, transient analysis is vital for capturing short-term dynamics and performance during periods of change or under time-sensitive conditions. In this dissertation, we leverage these perspectives to capture the communication system's behavior over finite time intervals, thereby enhancing our ability to forecast performance accurately and optimize the system.

Active Queue Management

Congestion control is critical in modern communication networks to ensure low latency and high throughput. One major challenge in this domain is *bufferbloat*, a phenomenon where excessively large buffers in routers or operating system network stacks cause high latency and jitter by holding packets for too long before they are transmitted [19]. Bufferbloat arises from traditional passive queueing methods, such as tail drop, where packets are simply buffered until the queue is full. This unregulated buildup can lead to long delays, reduced responsiveness in interactive applications, and overall degraded network performance [19].

Active queue management (AQM) was introduced to mitigate the effects of bufferbloat by proactively managing queues [20]. Unlike passive queueing—where packets are held until buffers overflow—AQM schemes continuously monitor the queue state and make dynamic decisions, such as preemptively dropping or marking packets, to prevent excessive queue buildup. This proactive approach helps to signal congestion early, thereby encouraging senders to reduce their transmission rates and maintain lower delays [20, 21].

The development of AQM algorithms can be traced back to the early 1990s, when researchers recognized that traditional tail drop techniques were insufficient to prevent congestion-related issues [19, 20]. Early schemes such as random early detection (RED) were designed to address these shortcomings by probabilistically dropping packets before the queue became fully congested [20]. Since then, a wide range of AQM algorithms have been implemented in various systems, including commercial routers and operating system networking drivers (for example, Linux networking stacks incorporate several AQM schemes within their queuing disciplines) [21].

Conventional AQM techniques, such as RED, use the current queue length or congestion level as input to probabilistically drop packets before the queue becomes congested [20]. By doing so, these schemes signal to senders to reduce their transmission rates, thereby maintaining lower delays and preventing severe congestion. However, such methods often require careful parameter tuning and may struggle under rapidly varying traffic conditions [19]. To address these challenges, more advanced schemes have emerged. One prominent example is CoDel (Controlled Delay), which, rather than relying solely on queue length, actively monitors packet delays and dynamically adjusts its dropping strategy to maintain a target delay [22, 23]. This adaptive approach makes CoDel particularly effective in managing congestion and reducing bufferbloat in networks with heterogeneous and time-varying traffic patterns [23].

1.2.2 5G Wireless Communication Systems

The evolution of cellular wireless networks—from the early analog systems of 1G through digital and data-centric 2G/3G/4G technologies—has set the stage for the transformative capabilities of 5G [24, 25]. The 3rd generation partnership

project (3GPP) has long played a central role in standardizing these networks, and its efforts in the 5G domain (notably with the advent of 3GPP release 15) marked a significant milestone [26]. This new generation promised not only enhanced mobile broadband but also introduced ultra-reliable low-latency communication (URLLC) and massive machine type communication (mMTC). These advancements were expected to meet the demands of both consumer-driven applications and emerging time-critical industrial and internet of things (IoT) services, where minimizing end-to-end delay is paramount.

A key enabler of 5G's performance is its redefined data plane architecture, which is organized around three principal components: the user equipment (UE), the next generation NodeB (gNB), and the core network. As illustrated in Figure 1.3, in the uplink, data packets originate at the UE and are transmitted over the radio interface to the gNB; from there, packets are forwarded through the core network's user plane function (UPF) toward their destination. Conversely, in the downlink the UPF first routes packets to the gNB, which then transmits them to the UE. The UE acts as the end device generating or consuming data, the gNB manages radio resource allocation, scheduling, and signal processing, and the core network undertakes critical tasks such as mobility management, security, and interconnection with external networks. This layered arrangement, while offering scalable and flexible service delivery, inherently introduces multiple stages where delay can be accumulated.

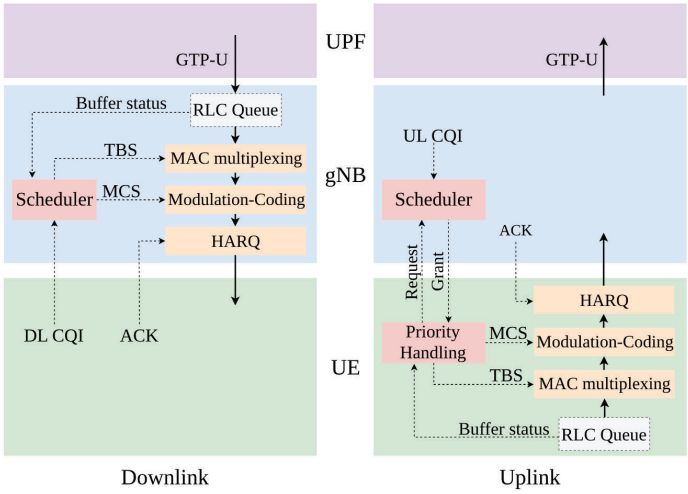


Figure 1.3: High-level view of data flows in a 5G network, illustrating how user-plane data moves between the UPF, gNB, and UE. Key processes such as buffering, scheduling, medium access control (MAC) multiplexing, modulation/coding, and hybrid automatic repeat request (HARQ) are highlighted to show their roles in end-to-end delay.

Queuing Mechanisms and Buffering

When a packet first enters the network, it is admitted into a buffering system where it awaits processing and transmission. This queuing stage is subject to inherently random delays, as the time a packet spends waiting depends on the instantaneous load and the current buffer occupancy. Variability in packet interarrival times and service times introduces stochastic behavior in the queuing delay. For instance, during periods of high traffic, packets may experience extended waiting times due to resource contention, while in lighter traffic conditions the delay is reduced. The randomness of these factors makes the queuing delay a critical stochastic component of the overall end-to-end latency [27].

Resource Scheduling

After buffering, a packet proceeds to the resource scheduling phase, a stage that introduces further stochastic delay through several intertwined processes. Initially, a scheduling handshake occurs between the UE and the gNB as control messages are exchanged to reserve transmission resources. The duration of these handshakes can vary significantly due to processing delays and signaling overhead.

With the advent of orthogonal frequency division multiplexing (OFDM) in 4G systems, the available transmission resources became organized into a time-frequency grid of resource blocks [28]. Once the scheduler detects that a queue is non-empty, it must decide both when and where to allocate these resource blocks, as well as how many of them to assign. This decision is influenced by the duplexing mode employed by the cellular network. In frequency division duplex (FDD) systems, uplink and downlink transmissions occur simultaneously on separate frequency bands, generally minimizing directional delay. In contrast, time division duplex (TDD) schemes divide transmission time into alternating uplink and downlink slots. Consequently, if a packet arrives during a slot allocated for the opposite transmission direction, it must wait for the appropriate slot to become available, thereby incurring additional delay.

Beyond the duplexing mode, network congestion plays a critical role. Even when sufficient resources exist, high traffic volumes force the scheduler to arbitrate among competing packets. Under such conditions, the allocation decision is subject to the inherent randomness of network load and scheduling policies. In some cases, the scheduler may even segment a packet—dividing it into multiple transmission blocks—with each block undergoing independent scheduling. These segmented transmissions introduce their own delays as each block competes for available resources. Additionally, the scheduler's allocation decision is influenced by the current modulation and coding scheme (MCS) index, which directly affects the efficiency of the link in terms of bits per resource block and its error performance. Collectively, these factors compound the overall transmission time through a series of stochastic delays [29].

Transmission Over the Wireless Link

Following resource scheduling, the packet is transmitted over the wireless channel—typically via the MAC and physical layers—where ensuring reliable delivery becomes the primary objective. In this stage, forward error correction techniques and redundancy mechanisms are employed by selecting an appropriate MCS index, which is central to balancing throughput with reliability [30].

The MCS index determines both the modulation format—such as QPSK, 16-QAM, or 64-QAM—and the coding rate used to encode the transmitted data. Its primary purpose is to maximize spectral efficiency while keeping the block error rate within acceptable limits. An optimal MCS index allows the system to achieve the highest possible data rate under prevailing channel quality metrics, such as signal-to-noise ratio and interference levels. However, the selected MCS index directly influences the probability of transmission errors. Higher MCS indices, characterized by denser modulation and lower redundancy, increase the likelihood of errors when channel conditions are suboptimal. In contrast, lower MCS indices employ more robust error-correcting codes and conservative modulation schemes, thereby reducing the error rate at the cost of throughput. As channel quality fluctuates, any degradation can lead to a spike in error rates, which in turn necessitates retransmissions and further extends the overall delay.

To cope with these challenges, the system continuously monitors channel conditions in a process known as link adaptation. This process relies on the channel quality indicator (CQI), which is periodically reported by the receiver to indicate the current state of the radio channel. Based on the reported CQI, the network adjusts the MCS index—raising it when the channel is favorable to boost throughput, and lowering it when the channel deteriorates to enhance reliability. Although this dynamic adjustment is essential for optimal performance, it also introduces additional variability into the transmission delay, as each adaptation cycle can alter the effective data rate [27, 29].

Enhancing transmission reliability further, the HARQ mechanism plays a crucial role. In HARQ, after a data block is transmitted, the receiver sends an acknowledgement (ACK) if the block is decoded correctly. If errors are detected, a negative acknowledgement (NACK) is issued. Rather than merely resending the same data, the transmitter provides incremental redundancy by transmitting additional parity bits. This approach allows the receiver to combine information from multiple transmission attempts until the data block is successfully decoded. The reliance on acknowledgements and the variability in the number of retransmissions required—driven by instantaneous channel conditions—introduce yet another layer of stochastic delay.

Together, the interplay of buffering, resource scheduling, link adaptation, and HARQ retransmissions underlines the dynamic and unpredictable nature of delay in the wireless channel. The cumulative effect of these mechanisms ensures that end-to-end delay is not solely determined by fixed network parameters, but is also heavily influenced by the fluctuating conditions of the wireless medium.

1.2.3 Probabilistic Prediction and Machine Learning

In many engineering and scientific applications, precise probabilistic prediction is essential to quantify uncertainty and to make informed decisions under stochastic conditions. In the context of network performance, risk management, and other time-sensitive domains, one is often interested not only in point estimates but also in the entire probability distribution of the quantity of interest. In this subsection, we provide a background on probabilistic prediction from a machine learning perspective. We first review the general problem of density estimation, introduce mixture density networks (MDNs) as a way to parameterize conditional densities, discuss how extreme value theory (EVT) can be used to better model the tails of these distributions, and finally illustrate how temporal models such as recurrent neural networks (RNNs) and Transformers are leveraged to capture sequential dependencies in the data.

Density Estimation

Density estimation involves inferring the underlying probability distribution of a random variable based on observed samples. In the unconditional case, given data points $\{z_i\}_{i=1}^{N}$ drawn from an unknown distribution, the objective is to estimate the probability distribution $p_Z(z)$ of the variable Z. In the conditional case, the aim is to estimate

$$p_{Z|X}(z \mid \mathbf{x}), \tag{1.11}$$

where $X \in \mathbb{R}^{d_x}$ is a vector of explanatory or contextual variables and $\mathbf{x}$ is a specific realization [31].

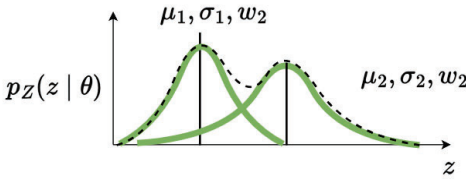


Figure 1.4: Gaussian mixture model example with 2 components and their parameters denoted by μ as mean, σ as variance, and w as mixture weight.

A popular data-driven strategy models the density using a parametric form $p_Z(z \mid \theta)$, where θ denotes the model parameters. A common approach within this framework is to represent the density as a mixture of K simpler component densities. For example, when these components are Gaussian, the model is referred to as a Gaussian mixture model (GMM) as shown in Figure 1.4. Formally, the mixture model is expressed as

$$p_Z(z \mid \theta) = \sum_{k=1}^{K} w_k \, p_Z^{(k)}(z \mid \theta_k), \tag{1.12}$$

where the mixing weights $w_k \geq 0$ satisfy $\sum_{k=1}^{K} w_k = 1$ and $p_Z^{(k)}(z \mid \theta_k)$ denotes the component densities parameterized by θ_k. Thus, the complete set of parameters is $\theta = \{w_k, \theta_k\}_{k=1}^{K}$. In the conditional context, both the mixing weights and the component parameters can depend on the context $\mathbf{x}$, leading to

$$p_{Z|X}(z \mid \mathbf{x}) \approx \sum_{k=1}^{K} w_k(\mathbf{x})\, p_Z^{(k)}\big(z \mid \theta_k(\mathbf{x})\big). \tag{1.13}$$

This formulation is particularly useful when the target distribution is multimodal or exhibits complex structures that a single parametric family cannot capture.

Other density estimation approaches include nonparametric techniques such as kernel density estimation (KDE) and histogram methods. KDE, for instance, estimates the density by centering a smooth kernel (e.g., Gaussian) at each observed data point and summing their contributions. Histograms, on the other hand, partition the data domain into bins and estimate the density based on the relative frequencies within each bin. More recently, normalizing flows have emerged as a powerful alternative; these methods construct complex distributions by applying a sequence of invertible transformations to a simple base distribution, thereby enabling exact likelihood evaluation and flexible modeling in high-dimensional spaces [31].

In parametric density estimation, the parameters θ are typically estimated via maximum likelihood. The log-likelihood function for the observed data $\mathcal{D} = \{z_i\}_{i=1}^{N}$ is given by

$$\mathcal{L}(\mathcal{D}) = \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} w_k\, p_Z^{(k)}(z_i \mid \theta_k) \right). \tag{1.14}$$

Direct maximization of $\mathcal{L}(\mathcal{D})$ is challenging due to the logarithm of a sum, which complicates the optimization landscape. Instead, one typically defines the loss function as the negative log-likelihood (NLL), $-\mathcal{L}(\mathcal{D})$, and employs iterative optimization techniques to minimize this loss, thereby effectively maximizing the likelihood.

Mixture Density Networks

While traditional density estimation methods often assume a fixed mapping from $\mathbf{x}$ to the parameters of the density, MDNs offer a flexible way to model such dependencies by coupling a neural network with a parametric density model. In an MDN, the mapping from the input space to the parameters of the density is given by a trainable neural network h_ω:

$$\theta(\mathbf{x}) = h_\omega(\mathbf{x}), \tag{1.15}$$

where the parameter vector $\theta(\mathbf{x})$ may comprise the mixture weights, component means, and variances (or other parameters depending on the chosen model). Training the MDN involves choosing the network parameters ω such that the NLL of the

dataset denoted by $\mathcal{D} = \{(z_i, \mathbf{x}_i)\}_{i=1}^{N}$ is minimized as

$$\mathcal{L}(\mathcal{D}) = -\sum_{i=1}^{N} \ln\Big(p_Z\big(z_i \mid h_\omega(\mathbf{x}_i)\big)\Big). \tag{1.16}$$

This procedure is equivalent to maximizing the conditional likelihood of the samples in $\mathcal{D}$ and ensures that the learned model faithfully captures the behavior of the underlying density. By coupling a flexible neural mapping with a parametric form, MDNs allow one to capture complex and possibly multimodal relationships between $\mathbf{x}$ and z [31].

Extreme Value Theory

EVT originated from practical needs in industries such as insurance, finance, hydrology, and environmental science, where understanding the risks of catastrophic events is paramount [32]. Early applications in insurance and economics—where predicting the occurrence of rare but costly events (e.g., large claims, market crashes, or natural disasters) was critical—motivated the development of a statistical framework dedicated to the study of extreme phenomena. Over time, addressing these practical challenges led to theoretical advances that now underpin EVT and provide a rigorous framework for analyzing the tail behavior of probability distributions.

At its core, EVT addresses the question of how the extreme values (or tails) of a distribution behave, analogous to how the classical law of large numbers and the central limit theorem describe the behavior of sums or averages. In essence, EVT can be thought of as a "law of large numbers for the extremes." Consider a sequence of i.i.d. random variables $Z_1, Z_2, \ldots, Z_n$ with cumulative distribution function $F(z)$. Let $M_n = \max\{Z_1, Z_2, \ldots, Z_n\}$ denote the maximum value in the sequence. Under suitable regularity conditions, there exist sequences of normalizing constants $\{a_n\}$ (with $a_n > 0$) and $\{b_n\}$ such that the distribution of the normalized maximum converges:

$$\frac{M_n - b_n}{a_n} \xrightarrow{d} G, \tag{1.17}$$

where G is a non-degenerate distribution known as the generalized extreme value distribution (GEV) distribution [32]. The GEV unifies three families—Gumbel, Fréchet, and Weibull distributions—which correspond to different types of tail behavior.

An alternative but complementary perspective focuses on values that exceed a high threshold rather than block maxima. In the threshold exceedance method, one considers the excess $Z - r$ for $Z > r$, where r is a high threshold. For a broad class of underlying distributions—including the normal, exponential, and lognormal distributions—the conditional distribution of these excesses converges, as r approaches the right endpoint of F, to the generalized pareto distribution

(GPD) [33]:

$$\lim_{r \to z_F} \Pr(Z - r \leq z \mid Z > r) = GPD(z \mid \beta, \xi, r), \quad z \geq 0, \tag{1.18}$$

with z_F denoting the right endpoint of F and GPD typically defined as

$$GPD(z \mid \beta, \xi, r) = \begin{cases} \frac{1}{\beta}\left(1 + \frac{\xi}{\beta}(z - r)\right)^{-1/\xi - 1}, & \xi \neq 0, \\ \frac{1}{\beta}\exp\left(-\frac{z-r}{\beta}\right), & \xi = 0, \end{cases} \tag{1.19}$$

where $z \geq r$, $\beta > 0$ is a scale parameter, and ξ is the tail index that characterizes the heaviness of the tail.

The block maxima and threshold exceedance methods offer different trade-offs. The block maxima approach, justified by the Fisher-Tippett-Gnedenko theorem, considers only one extreme observation per block (e.g., the maximum in each time period). This method tends to yield lower bias but high variance in case the block size is small and high bias and low variance in case of too large block sizes. Conversely, the threshold exceedance method uses all data points exceeding a predefined high threshold, which can reduce variance by decreasing the threshold; however, it introduces potential bias if the threshold is not set sufficiently high. Balancing this variance–bias trade-off is a central challenge in the practical application of EVT [32].

In summary, EVT provides a rigorous method for modeling the behavior of the extremes by offering two distinct yet complementary approaches: the GEV distribution for block maxima and the GPD for threshold exceedances. These theoretical insights are crucial for accurately quantifying tail risks in various applications, ranging from insurance and finance to environmental and engineering systems.

Machine Learning with Deep Neural Networks

Neural networks, also known as artificial neural networks, are computational models loosely inspired by the interconnected structure of neurons in the brain. Early research in this field centered around shallow architectures, such as Rosenblatt's perceptron in the late 1950s, which could perform simple classification tasks but struggled with more complex problems. Over time, it was realized that adding additional layers—leading to so-called *deep* neural networks—could dramatically increase representational capacity, allowing these models to learn hierarchical features from data rather than relying on hand-crafted features.

Early deep neural networks were predominantly feed-forward architectures, where the input vector is processed through successive layers to produce an output. Training these networks involves defining a loss function—such as the root mean square error (RMSE) or cross-entropy, or the NLL for probabilistic tasks—and minimizing it via backpropagation. As networks grew in scale and complexity, researchers discovered that careful hyperparameter tuning and more sophisticated optimization approaches were key to stable training. Adaptive optimizers like

Adam [34] have become standard due to their ability to adjust learning rates on a per-parameter basis.

In addition to proper weight initialization and loss minimization, several training techniques are crucial for robust performance:

- **Normalization and Standardization:** Scaling input features and applying methods like batch normalization [35] help stabilize the training process.
- **Residual Connections:** Introduced to alleviate vanishing gradients, residual (skip) connections allow gradients to bypass certain layers, facilitating the training of very deep networks [36].
- **Dropout and Noise Regularization:** Dropout [37] randomly deactivates neurons during training, while additive Gaussian noise—another form of noise regularization—further improves generalization, particularly in probabilistic estimation tasks where NLL loss is used.

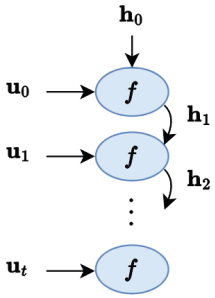


Figure 1.5: An unrolled recurrent neural network

Building on these feed-forward models, RNNs were introduced to handle sequential data by maintaining a hidden state that evolves over time [38]. RNNs process inputs in an autoregressive fashion: at each time step t, the network updates its hidden state h_t based on the current input vector $\mathbf{u}_t$ and the previous hidden state h_{t-1}:

$$h_t = f\big(\mathbf{u}_t, h_{t-1}\big). \tag{1.20}$$

This recursive formulation enables the network to generate predictions sequentially. During training, the network is unrolled over a fixed sequence length and trained via backpropagation through time. In multi-step prediction scenarios, once the model processes an initial sequence of tokens, it generates future outputs in an autoregressive manner—typically using its own previous predictions as inputs when true future data is unavailable, and padding tokens may be employed to maintain consistent input dimensions.

Standard RNNs, however, often struggle to capture long-term dependencies due to vanishing gradients. Long short-term memory (LSTM) networks [39] address this limitation by introducing an additional cell state c_t alongside the hidden state h_t. LSTMs employ gating mechanisms—namely, the input, forget, and output gates—to regulate the flow of information and update both h_t and c_t. In practice, stacking multiple RNN or LSTM layers (i.e., using a multilayer architecture) is common to enhance the model's capacity; each successive layer captures increasingly abstract temporal features, thereby improving predictive performance.

Moreover, modern sequence modeling typically begins with the tokenization of raw inputs into fixed-dimensional representations. In practical scenarios, raw data (e.g., high-dimensional network states or packet contexts) are often vast and heterogeneous. A dedicated, trainable embedding network can learn to extract the most relevant information from each time step's input vector and compress it into a compact token. This tokenization not only reduces dimensionality but also facilitates the subsequent capture of temporal dependencies. Such techniques are widely employed in transfer learning and form the backbone of large language models (see, e.g., [40, 41]).

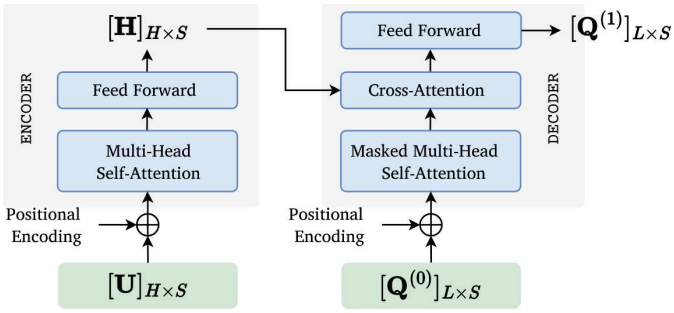


Figure 1.6: A transformer-based sequence to sequence mapping.

More recently, the Transformer architecture [42] has revolutionized sequential modeling by forgoing recurrence entirely in favor of self-attention mechanisms. As illustrated in Figure 1.6, in a Transformer, the process begins in the encoder: the tokenized input $\mathbf{U} \in \mathbb{R}^{H\times S}$ is first augmented with positional encodings to retain order information, then passed through a series of encoder layers (where S denotes the hidden size, and H denotes the length of historical observations). Each encoder layer consists of two primary sub-modules:

1. **Self-Attention:** This module computes relationships between all pairs of input tokens simultaneously, enabling the model to weigh the importance of each token relative to the others.

2. **Feed-Forward Network:** A position-wise network refines the output of the

self-attention module. Both sub-modules are wrapped with residual connections and layer normalization to ensure stable training.

The encoder outputs a contextual representation $\mathbf{H} \in \mathbb{R}^{H \times S}$ that encapsulates the full sequence's dependencies.

Then, for the prediction task, mainly the transformer's decoder component is involved. The decoder operates on a separate target sequence $\mathbf{Q} \in \mathbb{R}^{L \times S}$, where L denotes the number of future time steps to predict. Similar to the encoder, positional encodings are added to $\mathbf{Q}$. The decoder then processes its input through three main sub-layers:

1. **Masked Self-Attention:** This layer allows the decoder to attend only to previously generated tokens by applying a causal mask, thereby preventing access to future positions.

2. **Cross-Attention:** In this module, the decoder's token representations attend to the encoder output $\mathbf{H}$, effectively integrating historical context with the prediction process.

3. **Feed-Forward Network:** A position-wise network further refines the decoder's representations.

The output from the final decoder layer is then projected, typically via a fully connected layer, to yield the desired parameter vectors for prediction. As demonstrated in the original Transformer paper [42], stacking multiple layers in both the encoder and decoder significantly enhances the model's ability to capture long-range dependencies and complex temporal patterns.

In summary, the progression from feed-forward networks to RNNs, LSTMs, and ultimately Transformers represents a gradual evolution in deep learning for sequential data. Beginning with tokenization to transform raw, high-dimensional inputs into compact embeddings, each subsequent architectural innovation addresses specific challenges—such as the need for autoregressive modeling in RNNs, the retention of long-term dependencies in LSTMs, and the global context modeling via self-attention in Transformers—culminating in highly effective models for tasks ranging from language processing to time-dependent probabilistic prediction.

CHAPTER **2**

Scope & Related Work

In this chapter, we begin by describing the specific scope and challenges that underpin this thesis in Section 2.1, we then review the relevant literature in Section 2.2, providing an overview of related works that have contributed to the foundation of this research.

2.1 Scope of the Thesis and Research Questions

Throughout this thesis, we examine time-sensitive applications that depend on the continuous delivery of periodic data packets, each of which must adhere to stringent end-to-end delay constraints. Consequently, we assume periodic packet arrivals in our analysis. From this perspective, and given the inherent variability of wireless communications and computation times, we argue that delay should be treated as a probabilistic measure rather than as a fixed value. The random fluctuations of the wireless medium and other unmonitored factors necessitate modeling delay as a distribution. This probabilistic view underpins our efforts to develop methods that accurately predict the likelihood of a given delay occurrence, thereby yielding delay violation probability (DVP) predictions with respect to the application's delay requirements. These predictions can then be integrated into network adaptation strategies or reported back to the application. Moreover, in ultra-reliable low-latency communication (URLLC) scenarios—where the required delay violation probabilities may be as low as 10^{-5}—the focus shifts from capturing average behavior to accurately predicting these rare but critical delay outliers. Then a big challenge lies in developing predictors that excel at characterizing the tail of the

delay distribution efficiently and in rigorously evaluating their performance and sensitivity.

With this overarching goal of probabilistic delay prediction in mind, we specify detailed research questions through a two-stage scoping strategy to achieve the thesis objectives outlined in Chapter 1. Initially, we utilize abstract queueing-theoretic models to investigate fundamental elements impacting delay prediction, allowing for an analytical and theoretical examination of the core issues. This controlled framework offers an established analytical basis for investigating core questions like predictability, developing foundational concepts, and rigorously testing their implications. It also provides a confined setting in which we can implement and refine early versions of delay prediction and control algorithms. Next, we transition to a real 5G environment, where we validate our methods under actual operating conditions. This shift uncovers additional complexities—particularly regarding efficient and scalable delay-focused data collection and measurement in 5G networks—which we address accordingly. Moving from abstract theoretical modeling to real-world experimentation is vital for demonstrating both the feasibility and the broader relevance of our strategies in dynamic wireless systems.

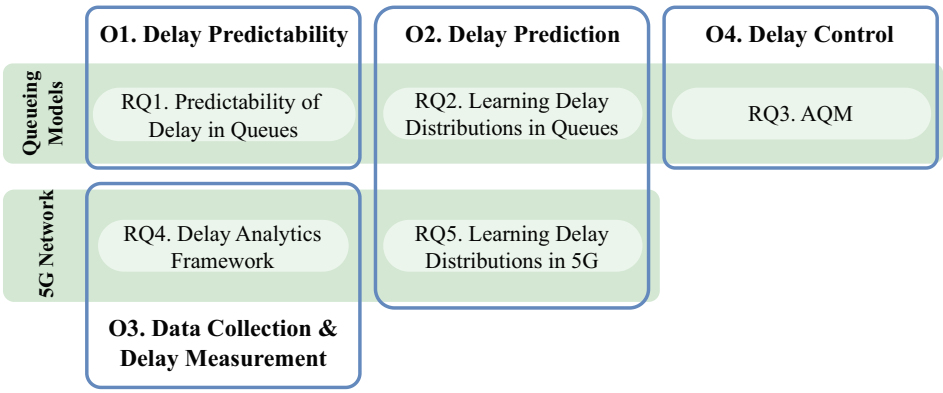


Figure 2.1: Overview of the scope of the thesis

In this thesis, we address seven research questions, as shown in Figure 2.1, each aligned with a specific objective within the overall scope:

- Theoretical Scope: Queueing Models
  - **Research Question 1.** From a probabilistic-performance perspective, how should delay predictability be formally defined for communication networks, how can this definition be instantiated within multi-hop Markovian queueing models, and in what ways do different levels of observability—and the gaps therein—govern the accuracy of end-to-end delay predictions? (Paper A)

  - **Research Question 2:** In tandem multi-hop queueing systems, how can data-driven, machine-learning techniques be leveraged to efficiently estimate the full end-to-end delay distribution—especially the probabilities of rare, extreme delays that reside in the distribution's tail? (Paper B)
  - **Research Question 3.** How can probabilistic delay predictions be integrated into Active Queue Management (AQM) policies for single-hop queues, and how do these enhanced policies perform when implemented and benchmarked under controlled conditions? (Paper C)

- Experimental Scope: 5G Networks
  - **Research Question 4.** How should the 5G protocol stack be instrumented to achieve full visibility into the end-to-end delay distribution, and how can the resulting measurements be analyzed to provide actionable insights for delay control? (Papers D and E)
  - **Research Question 5.** How can machine learning techniques be used to accurately estimate end-to-end delay probabilities in operational 5G networks—including rare, extreme delays—and to forecast the full delay distribution at specified future time horizons? (Papers F and G)

These research questions guide our exploration from theoretical modeling through practical experimentation, helping ensure our methods are both scientifically sound and practically relevant. This structured approach naturally leads into the subsequent section on related works, where we review the existing literature on predictability, delay prediction, and active queue management for wireless networks, thereby positioning our research within the broader academic landscape.

2.2 Related Work

In this section, we survey the literature that forms the foundation for our research. We organize the discussion into four primary categories: (i) the predictability in communication networks (RQ1), (ii) delay prediction for wireless networks (RQ2 and RQ5), (iii) active queue management for delay-sensitive applications (RQ3), and (iv) data collection and delay measurement in 5G (RQ4). Together, these works inform the development of our framework for end-to-end delay prediction and system optimization in time-sensitive communication systems.

2.2.1 Predictability in Communication Networks

The study of predictability—understanding the limits of how accurately we can forecast the future behavior of a system—has a rich history across various disciplines. Researchers have developed a range of measures to assess the predictability

of complex dynamic systems, with a common focus on predicting critical performance metrics, whether single- or multi-dimensional. These efforts can be categorized based on (i) the comprehensiveness of their predictability definition, (ii) whether they incorporate side information (contextual data) in addition to the target metric itself, and (iii) whether the predictability analysis is empirical or model-driven.

A foundational approach involves predicting a single, univariate metric without side information. Common techniques in this category include autocorrelation, widely used in finance to assess how a time series correlates with its lagged versions [43], and permutation entropy (PE) [44–46]. PE quantifies the complexity of a time series by examining the diversity and frequency of ordinal patterns, with higher entropy indicating lower predictability. PE has found applications in diverse fields, including epidemiology [47] and ecology [46]. Abeliuk et al. [48] further explored the interplay between predictability and sampling in partially observed systems, evaluating measures like PE and autocorrelation across various domains.

Another approach focuses on determining lower bounds on prediction error. Fano's inequality, combined with empirically determined entropy, is often used to establish these bounds. For instance, Li et al. [49] applied this method to study the predictability of urban vehicular location and dwell time. Information-theoretic measures, such as random entropy, uncorrelated entropy, and conditional entropy, have also been employed to quantify predictability, notably in studies of human mobility and communication patterns [50, 51].

More sophisticated approaches incorporate side information or covariates into the predictability model. This often involves using relative entropy, conditional entropy, or mutual information between the side information and the target metric. Examples include Bialek et al.'s definition of predictive information as the mutual information between past and future states [52], Haven et al.'s use of Gaussian ensemble prediction and relative entropy [53], and Li et al.'s application of conditional entropy to traffic forecasting [54]. Fang and Lee [55] explored predictability in machine learning regression problems, deriving bounds on conditional entropy between features and labels.

In atmospheric science, where probabilistic forecasting is paramount, information-theoretic definitions of predictability incorporating side information are prevalent [56, 57]. DelSole et al. define predictability as the divergence between forecast and climatological (marginal) distributions, effectively using the marginal distribution as a baseline representing prediction without side information [58].

Within networking and communication systems, Ding et al. [59] used Fano's inequality to investigate the predictability of radio spectrum state dynamics, finding high predictability with implications for cognitive radio and 5G spectrum sharing. Sihai et al. [60] analyzed voice traffic predictability using entropy, establishing bounds on prediction accuracy and exploring the effectiveness of N-order Markov models. Jing et al. [61] studied structural predictability in directed networks, linking it to controllability and defining it empirically through link prediction accuracy.

While QoS prediction in communication systems has received increasing attention, a comprehensive analysis of predictability in this context is, to our knowledge, lacking. This thesis addresses this crucial gap. We adopt a rigorous, probabilistic definition of predictability inspired by atmospheric science [58], comparing the conditional distribution of performance (given side information) to its marginal distribution with total variation distance. This approach allows us to model imperfect observability and is applicable to a wide range of dynamic systems.

2.2.2 Delay Prediction in Wireless Networks

Delay prediction in communication networks has been tackled through both analytical and data-driven approaches. Analytical methods—most notably those based on stochastic network calculus—are rooted in queuing analysis and aim to capture the queuing delay that arises due to random service times or the inherent variability of wireless links. By modeling the accumulation of delays as data packets traverse through network queues, these methods derive probabilistic bounds on the end-to-end delay. This approach, while effective in highlighting how randomness in service processes impacts overall network performance, often relies on simplifying assumptions that may restrict its applicability in more dynamic and complex real-world scenarios.

A number of works have focused on developing analytical models using these approaches. For example, Al-Zubaidy et al. [13] present a network-layer performance analysis for multihop fading channels using stochastic network calculus. Their model assumes a stationary environment under Rayleigh block fading with uncorrelated services only, thereby simplifying the analysis while limiting its applicability to more complex or dynamic conditions. Building on this, Champati et al. [62] extend the framework to encompass both stationary and transient analyses, still within the context of Rayleigh block fading and uncorrelated services. The inclusion of transient analysis in their work is particularly noteworthy, as it provides insights into the dynamic behavior of wireless networks over time. More recently, Coll-Perales et al. [63] propose a model-based probabilistic approach for end-to-end vehicle-to-everything (V2X) latency modeling in 5G networks. Unlike the earlier works that rely strictly on stochastic network calculus, their approach adopts a more generalized probabilistic model.

Together, these analytical models illustrate both the strengths and limitations of using stochastic network calculus for delay prediction. While the simplifying assumptions—such as stationarity, Rayleigh block fading, and uncorrelated service processes—aid in tractable analysis, they also highlight the challenges in extending these results to scenarios where network conditions exhibit significant temporal or spatial variability.

On the other hand, data-driven approaches offer greater flexibility in modeling delay prediction, particularly when leveraging deep learning to capture the complex dynamics of wireless communication systems. These methods learn the intricate relationships between delay and various influencing factors directly from measure-

ment data, and when designed to incorporate temporal dependencies, they can effectively capture trends from historical observations. One of the main advantages of these approaches is their broad design flexibility. In addition, when evaluating related work, key distinguishing factors include:

- Output Representation: Some frameworks provide simple point estimates of delay, while others yield a full probabilistic characterization of the delay distribution.

- Density Estimation Methods: Models may adopt parametric techniques (e.g., Gaussian mixture models) or non-parametric approaches (e.g., histograms) to estimate the underlying delay distribution.

- Conditional Versus Non-Conditional Prediction: Certain methods condition predictions on side information or historical data, whereas others predict delay in a non-conditional fashion.

- Temporal Modeling: Approaches vary from non-temporal models (e.g., feed-forward deep neural networks) to temporal ones (e.g., long short-term memory (LSTM) or Transformer architectures) that account for sequential dependencies.

- Evaluation Setups: Studies differ in their validation environments, ranging from queuing models and simulation platforms to real-world experiments using 4G, 5G, or WiFi networks.

Among the data-driven techniques developed within a queuing-theoretic framework, Ibrahim et al. [64] introduced a analytical data-driven approach that leverages delay history to predict the expected waiting times in multi-server queues, while Senderovich et al. [65] evaluated a variety of predictors—including regression-based methods derived from delay history and queue snapshots. However, these investigations primarily concentrated on average delay metrics rather than capturing the complete delay distribution. In contrast, Raeis et al. [66] advanced the field by utilizing mixture density networks (MDNs) to forecast the entire distribution of end-to-end delay conditioned on queue backlogs.

On less abstract studies, simulation-based studies also play a vital role in evaluating data-driven delay prediction techniques. For example, Moreira et al. [67] compared different methods—including multi-layer perceptrons (MLPs), random forests, and autoregressive integrated moving average (ARIMA)—for point delay estimates in a simulated 5G vehicular scenario. Similarly, Barmpounakis et al. [68] utilized NS3 simulations for packet latency prediction in a V2X context by applying LSTM networks, while Dang et al. [69] applied bidirectional LSTMs to estimate point delays and introduced optimization policies to improve performance in a simulated 5G environment.

Although simulation environments provide a controlled setting to develop and test these methods, their reliance on simplifying assumptions can limit their practical validity. Real-world data is indispensable because it captures the full complexity and unpredictability of wireless networks, ensuring that predictive models remain robust and relevant under operational conditions.

A number of studies have been conducted on real wireless network data, where many works focus on point estimates or the estimation of specific quantiles of delay [70–73]. Notably, Rao et al. [73] concentrated on one-way delay prediction using domain adaptation techniques to improve generalizability across unseen user equipment, training neural networks on a real 5G millimeter-wave (mmWave) testbed. In another study, Palaios et al. [74] collected highway data from connected vehicles in a test LTE network to measure throughput and latency, employing models such as decision trees and MLPs for point delay prediction. Other works have addressed temporal dependencies in delay; for instance, in [75], an LSTM-integrated framework was proposed for point delay estimation in both fixed and mobile scenarios within LTE networks. More recently, attention-based autoregressive models—such as Transformers—have been used for quality of service (QoS) prediction, with studies like [76] and [77] demonstrating the superiority of Transformers over traditional approaches (e.g., LSTM and autoregressive integrated moving average (ARIMA)) for predicting wireless channel characteristics on real datasets.

Despite these advances, point estimates are often insufficient for many delay-sensitive applications that require a complete probabilistic characterization of delay. Methods capable of capturing the full stochastic nature of delay provide richer information that is more useful for ensuring robust performance in time-sensitive applications. Some works have addressed probabilistic delay prediction on real wireless data using non-conditional approaches. For example, Volos et al. [78] employed mixture models on real LTE delay measurements, demonstrating that mixtures of lognormals and extreme value theory models (such as the generalized pareto distribution (GPD)) can more effectively model rare latency events. Similarly, Fadhil et al. [79] investigated the latency profile in 5G networks using Gaussian mixture models, examining how increasing the number of mixture centers influences fit accuracy.

Further refinement of probabilistic delay prediction is achieved when side information is incorporated. Flinta et al. [80] used random forests and a histogram-based approach to condition delay distribution predictions on contextual variables such as position, time, radio channel, and signal power in internet of things (IoT) networks. In a related study targeting cloud environments, Samani et al. [81] combined deep neural networks with Gaussian mixture models in the framework of MDNs to predict service metrics from conditions such as central processing unit (CPU) utilization.

Temporal probabilistic prediction on real-world data is still a relatively under-explored area. Skocaj et al. [82] leveraged recurrent neural networks (RNNs) and LSTMs on mobile network operator data to derive conditional delay probability density functions that account for dependencies on network and traffic conditions.

Nonetheless, their method is confined to single-step predictions, and its sensitivity and efficiency were not rigorously assessed, positioning it primarily as a proof of concept.

A significant gap in the current literature (and one that this thesis addresses) is the development of multistep probabilistic forecasting, which is crucial for proactive network operations. While many approaches predict delay for only a single future time step, autoregressive models that capture temporal dependencies across multiple time steps can extend the utility of predictions, enabling network operators to plan for a longer future horizon. Moreover, most existing probabilistic prediction methods have concentrated on the body of the delay distribution, with limited focus on the tail. This is a critical shortcoming for URLLC scenarios, where accurately capturing the tail behavior of the delay distribution is essential.

Another observation from the literature is the prevalent reliance on manual feature selection for data encoding and embedding. Such approaches can impede scalability. In contrast, our approach leverages tokenizer architectures to automatically learn relevant features and discard irrelevant data, thereby enhancing scalability and robustness in QoS prediction.

In summary, while data-driven delay prediction frameworks have advanced considerably—from simulation-based evaluations to models trained on real-world data—there remains a need for methods that offer multistep probabilistic forecasts with accurate tail characterization and scalable data encoding. These are precisely the challenges that this thesis seeks to address.

2.2.3 Delay Control with Active Queue Management

Active queue management (AQM) aims to mitigate the bufferbloat problem, where excessively large queues in network devices lead to high latency and jitter. Traditional AQM schemes, such as proportional integral enhanced (PIE) [83] and controlled delay (CoDel) [84], attempt to control queuing latency by either limiting the average delay (as in PIE) or tracking the minimum queue sojourn time (as in CoDel). However, these schemes are typically designed for wired networks, where link conditions are relatively stable. In wireless links, rapid fluctuations in channel quality, variable link conditions, and intermittent connectivity often undermine the assumptions underlying traditional AQM, making them less effective at managing delay.

More recent approaches leverage deep reinforcement learning (DRL) [85] to improve AQM performance. While these methods frequently yield superior results compared to conventional techniques, they typically depend on carefully tuned reward functions. This reliance on fine-tuning can limit their generalizability across different applications and network conditions.

Another line of research focuses on minimizing end-to-end latency. Liu et al. [86] proposed a method that integrates congestion control with queue management to reduce jitter and promote more deterministic latency. Similarly, Kar et al. [87] formulated a semi-Markov decision process to derive an optimal packet dropping

policy based on probabilistic models of flow rates and service times. However, the assumption of exponentially distributed service times in their work may not always hold in practice, potentially limiting the applicability of their findings.

Efforts addressing the specific challenges of wireless networks have further expanded the AQM landscape. Irazabal et al. [88] investigate bufferbloat in the radio access network (RAN) and propose an AQM algorithm that estimates available bandwidth using a history of previous bandwidth measurements or resource allocations over a defined window. This tailored approach mitigates the bufferbloat problem in wireless environments by accounting for the dynamic nature of resource availability. Building on this work, Stoltidis et al. [89] implemented Irazabal et al.'s proposals in a cloud RAN scenario, further demonstrating the practical benefits of the proposed AQM strategy.

This thesis contributes to AQM research by demonstrating how imperfect yet informative probabilistic delay predictions can be effectively integrated into network management. We formulate a problem that shows how DVP predictors—even with some error—can significantly improve network control decisions. Our approach integrates DVP predictions to assess and enhance end-to-end delay performance, offering a practical method for managing delay in real-world systems. This stands in contrast to techniques that rely on strict distributional assumptions or require extensive reward function tuning.

2.2.4 Data Collection and Delay Measurement in 5G

5G incorporates standard mechanisms for recording, analyzing, and exposing key-performance indicators (KPIs) including packet-delay metrics. 3GPP defines the network data analytics function (NWDAF) as the entity responsible for collecting network data and applying analytics or machine-learning techniques to extract actionable insight. NWDAF ingests two complementary streams: (i) periodic performance counters that capture statistics such as user-plane latency across the RAN, transport, and core; and (ii) observe-event reports that convey real-time context (e.g., an ongoing handover). All counters are time-stamped and stored so they can be correlated with events before analytics or ML models produce forecasts or alerts. The resulting information is exposed through the network exposure function (NEF), which acts as a secure interface for external parties. Within the core, an application function (AF) can also consume NWDAF outputs to optimise application behaviour [90–92]. A substantial body of research has employed this structure to tackle performance-related prediction and analysis tasks, as well as proactive adaptation challenges [93–97].

Although NWDAF and NEF provide the architectural hooks for latency analytics, the way in which the 5G RAN and transport stack should be instrumented to supply the most useful latency-related raw data is still an open question. We turn to recent research that probes the problem from complementary angles to understand which measurements are actually needed for end-to-end (E2E) delay.

In this context, Scano et al. [98] attach a P4-INT header to latency-sensitive packets and propose to insert timestamps at the UE and at every hop—including the wireless link, so each packet carries its full end-to-end delay trace. The method, however, increases packet overhead, and the limited header space restricts how many layer-specific timestamps or extra metrics (e.g., channel quality) can be inserted, which reduces overall efficiency. Larrabeiti et al. [99] review how established operation and maintenance (OAM) tools from high-performance Ethernet networks can be applied to obtain delay measurements in sliced 5G networks. Their guidance helps operators set strict latency budgets and rapidly identify bottlenecks across the RAN, fronthaul/backhaul and core. The work, however, does not address how to collect or analyse RAN-specific delay context and insights. Wan et al. [100] introduce NR-Scope, a passive telemetry framework that decodes 5G RAN's control-channel messages to stream millisecond-scale metrics such as queue depth, scheduling latency, retransmission counts, to external applications without modifying gNB software or user traffic. They show that this visibility enables faster adaptation than end-to-end probes, yet it offers no insight beyond the RAN and does not provide per-packet, end-to-end delay traces.

Finally, Ronteix-Jacquet et al. [101] present LatSeq, which logs timestamps and related context at every LTE RAN layer each time a packet or its segment passes, then reconstructs the packet's full delay path offline to reveal the individual contributors to total latency. The design targets minimal overhead on the RAN stack. However, capturing and processing such detailed traces in high rates is resource-intensive. Their prototype is LTE-based and relies on basic post-processing, but it motivated us to extend the same idea with richer analytics and pipeline to 5G. Moreover, a systematic breakup of end-to-end latency and clear quantification of each part's role in missing delay budgets remain largely unexplored. Likewise, a framework capable of gathering these measurements and turning them into actionable insights and optimization guidance has not yet been developed—an absence this thesis aims to fill.

CHAPTER **3**

Key Contributions & Results

In this chapter, we present our key contributions to the literature on end-to-end delay prediction in wireless, time-sensitive applications. These contributions—together with supporting results—are introduced in three primary sections. However, before describing these in detail, we first discuss the unified system model and overarching approaches that underpin the analyses in the rest of this chapter in Section 3.1.

In Section 3.2, we introduce the definition of predictability tailored for forecasting delay in communication networks. We then analyze both single-hop and multihop scenarios, applying our framework to networks whose conditions evolve under Markovian dynamics such as Geo/Geo/1 queues. This section summarizes our studies to address Research Question 1.

Next, Section 3.3 presents a data-driven framework for delay prediction and AQM in queuing systems. We begin by describing a novel tail-optimized delay prediction method that accurately captures the tail of the delay distribution addressing Research Question 2. We then show how DVP predictions can be integrated into AQM, illustrating how even imperfect predictors can be leveraged to enhance network performance and meet stringent application-level delay requirements as the contributions towards Research Question 3.

Finally, Section 3.4 extends our investigation to 5G wireless systems. Here, we outline a comprehensive 5G delay analytics framework and discuss the implementation of a testbed that enables robust delay measurement and data collection as to address Research Question 4. Building on this foundation, we develop advanced tail-centric and temporal models to capture the time-varying nature of the wireless link, thereby facilitating accurate and proactive delay prediction in real-world 5G environments that are targeting Research Question 5.

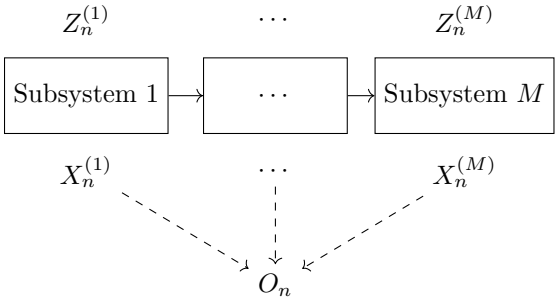


Figure 3.1: Multi-hop communication system model with observable measures being conditions.

3.1 General System Model and Methods

In this thesis we employ a probabilistic formulation for forecasting end-to-end packet delay in single or multihop networks, supporting time-sensitive applications. These applications often rely on the periodic exchange of fixed-size packets and mandate timely packet delivery. We consider a scenario with periodic traffic, where packets $n \in \{1, \ldots, N_s\}$ of a constant size B are transmitted at regular intervals to a system as shown in Figure 3.1. Let T_n denote the generation time of packet n. Due to the inherent stochasticity of the subsystems (either communication or computation links), we model the end-to-end latency of packet n as a random variable $Z_n \in \mathbb{R}_+$. Crucially, these latencies can exhibit serial correlation.

We consider a communication network that operates either as a single-hop or a multihop system. In the multihop case, the end-to-end delay experienced by a packet is modeled as the sum of the delays incurred at each hop. For a packet indexed by n, the overall delay is given by:

$$Z_n = \sum_{m=1}^{M} Z_n^{(m)}, \tag{3.1}$$

where $Z_n^{(m)}$ denotes the delay at the m-th hop and M is the total number of hops.

The network state at time n is represented by the random variable X_n. In a single-hop scenario, X_n captures the state of that single communication link (e.g., queue lengths, channel conditions, etc). In a multihop scenario, the state is naturally extended to a vector:

$$X_n = \left(X_n^{(1)}, X_n^{(2)}, \ldots, X_n^{(M)}\right), \tag{3.2}$$

where each $X_n^{(m)}$ represents the state at the m-th hop.

In addition, we introduce an observation model to describe the measurable aspects of the system. This step is crucial in modeling defects in the monitoring

system. Every delay prediction comprises a monitoring system to observe and transport the conditions or delay-related measurements to the prediction point and to the training and prediction processes themselves. The observation at time n is defined as:

$$O_n = \phi(X_n), \tag{3.3}$$

with the ideal scenario being $O_n = X_n$.

To characterize different levels of prediction, we define three fundamental delay distributions:

1. **Marginal (or Prior) Delay Distribution:**

$$\Pr(Z_n) \tag{3.4}$$

   This distribution represents the unconditional behavior of the delay without any side information.

2. **Posterior Distribution:** The posterior distribution is defined as the delay distribution when the observations become available. It can be expressed in two versions:

   a) Based on the current observation:

$$\Pr(Z_n \mid O_n = o_n). \tag{3.5}$$

   b) Based on the entire observation history:

$$\Pr(Z_n \mid O_{0:n} = o_{0:n}), \tag{3.6}$$

   where $O_{0:n} = \{O_0, O_1, \ldots, O_n\}$.

3. **Forecast Distribution:** Leveraging temporal information from the observation history, the forecast distribution is defined as:

$$\Pr(Z_{n+L} \mid O_{0:n} = o_{0:n}), \tag{3.7}$$

   where L is the forecast lead time.

In this thesis, we are mainly concerned with developing and analyzing methods and algorithms that accurately estimate the delay distributions defined above in a data-driven manner as required in Research Questions 2 and 5. Before presenting these methods, we first introduce our definition of predictability presented in Paper A based on the models described above, as part of contributions to Research Question 1.

3.1.1 Predictability Definition

We define predictability based on the difference between two distributions: the forecast distribution (conditioned on available observations) and the marginal distribution (unconditioned on observations). A system is deemed unpredictable when these two distributions are indistinguishable, implying that observations provide no advantage in forecasting.

To quantify predictability in our system, we measure their discrepancy using the total variation distance:

$$D_n(L) = \| \Pr(Z_{n+L} \mid O_{0:n} = o_{0:n}) - \Pr(Z_{n+L}) \|_{\mathrm{TV}}, \tag{3.8}$$

where L is the forecast lead time. In short, a high value of $D_n(L)$ indicates that the available observations significantly change our idea about future delay, implying that the system is highly predictable given the current monitoring data. Conversely, a low value suggests that the observations provide little additional information, thereby limiting the accuracy of delay predictions.

This definition not only measures the utility of past observations for the forecasts, but also helps us identify specific states of the network in which performance is inherently unpredictable. When $D_n(L)$ remains small for certain network states, observations offer little benefit, meaning additional monitoring or more sophisticated predictive tools may yield negligible gains. In such cases, it might be more efficient to redirect resources elsewhere rather than investing heavily in prediction. Conversely, when $D_n(L)$ is large, the system's state at that time is highly predictive, warranting more intensive monitoring efforts.

3.1.2 Prediction Methods

To estimate delay distributions, we employ a machine learning technique known as mixture density networks (MDN). Using this framework, the delay is modeled by a parametric probability density function $p_Z(z \mid \theta)$ (for example, a Gaussian mixture model), where the parameter vector θ of dimension V is learned by a neural network. This network takes as input features derived from network condition observations (for example O_n or X_n) to predict the complete delay distribution.

For estimating the prior delay distribution $\Pr(Z_n)$, the parameter vector θ can be estimated directly, without invoking the neural network component of an MDN. In this setting, the optimizer determines θ by minimizing the negative log-likelihood over a training set composed solely of independent and identically distributed (i.i.d.) delay samples.

To estimate the posterior distribution $\Pr(Z_n \mid X_n = \mathbf{x}_n)$, we use a complete MDN with the neural network h_ω to map the observed features $\mathbf{x}_n$ to the corresponding parameter vector, i.e., $\theta_n = h_\omega(\mathbf{x}_n)$. For training, a collection of i.i.d. samples is needed in the form:

$$\mathcal{D} = \{(\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), \dots\}, \tag{3.9}$$

which allows us to predict a delay distribution for each observation $\mathbf{x}_n$ and compute the corresponding log-likelihood based on the observed delay z_n. The training objective is to minimize the negative log-likelihood (NLL) over the dataset, updating the network parameters ω via standard stochastic gradient descent (SGD). Formally, for a dataset of N_D samples, the NLL which is defined as the loss value is derived via:

$$\mathcal{L}(\mathcal{D}) = -\sum_{n=1}^{N_D} \ln\Big[p_Z\big(z_n \mid \theta = h_\omega(\mathbf{x}_n)\big)\Big]. \tag{3.10}$$

This joint parameterization enables conditional inference, allowing the predictor to generate distinct delay distributions for different network states. In implementation, we add a slight Gaussian jitter to each delay sample z_n so that the model can cope with the data's discrete nature. Because the neural network's output layer must produce the mixture's means, variances, and weights, we employ tailored activation functions—linear for the means, softplus for the variances, and `softmax` for the weights. Finally, normalising the delay values to zero mean and unit variance before training delivers a marked improvement in predictive accuracy.

We adapt the basic MDN concept into a set of customised architectures, each crafted for a specific delay-prediction objective relevant to time-critical wireless networks. The forthcoming subsections present these variants in detail and explain the design considerations behind them.

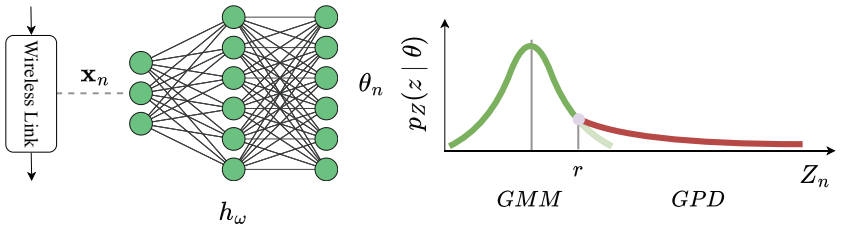


Figure 3.2: MDN-based delay prediction approach with accurate tail customization

Prediction with Accurate Tail

For the choice of the parametric density $p_Z(z \mid \theta)$, traditional Gaussian mixture models (GMMs) are commonly employed within MDNs. However, their performance often degrades in the tail regions of the delay distribution. To address this limitation, inspired by [102, 103], we proposed to incorporate extreme value theory (EVT) models into our MDN framework, enabling more accurate modeling of rare events in the tail.

Specifically, we use the GPD to model the tail behavior. The GPD is well-suited for characterizing exceedances above a specified threshold r and its probability

density function (PDF) is given by:

$$GPD(z \mid \beta, \xi, r) = \begin{cases} \frac{1}{\beta}\left(1 + \xi\frac{z-r}{\beta}\right)^{-1/\xi-1}, & \xi \neq 0, \\ \frac{1}{\beta}\exp\left(-\frac{z-r}{\beta}\right), & \xi = 0, \end{cases} \tag{3.11}$$

where β is the scale parameter, ξ is the tail index, and r is the tail threshold. To seamlessly combine the GMM and GPD, we define the overall parametric density as:

$$p_Z(z \mid \theta) = \begin{cases} f_Z(z \mid \theta_{\text{gmm}}), & z \leq r, \\ \left[1 - F_Z(u \mid \theta_{\text{gmm}})\right] GPD(z \mid \beta, \xi, r), & z > r, \end{cases} \tag{3.12}$$

where $f_Z(z \mid \theta_{\text{gmm}})$ and $F_Z(z \mid \theta_{\text{gmm}})$ denote the parametric PDF and cumulative density function (CDF) of the GMM component, respectively, and $\theta = \{\theta_{\text{gmm}}, \beta, \xi, r\}$ comprises the full set of parameters. This hybrid model is called extreme value mixture model (EVM) in this thesis. An overview of the non-temporal approach is illustrated in Figure 3.2 for a better understanding.

Temporal Prediction

In the case where we aim to estimate the forecast distribution (or the posterior distribution conditioned on a sequence of observations), we begin by limiting our historical context to the most recent H packets and targeting predictions for the next L packets. Our goal is to approximate

$$\Pr\Big(Z_{n+l} \mid X_{n-H+1:n} = \mathbf{x}_{n-H+1:n}\Big) \tag{3.13}$$

for each $l \in \{0, \ldots, L-1\}$, where $\mathbf{x}_{n-H+1:n}$ denote the past H realizations of the observed network conditions.

In this scenario, the sequence of observations can uncover trends and time-based correlations that forecast future delays. Consequently, our method must be designed to capture these temporal dependencies and connect $\mathbf{x}_{n-H+1:n}$ to the parameters of distributions of upcoming delays. As illustrated in Figure 3.3, we first map each observation $\mathbf{x}_n$ to a fixed-dimensional token $\mathbf{u}_n \in \mathbb{R}^S$ using a learnable embedding function such as an MLP. Next, we gather the tokens $\{\mathbf{u}_{n-H+1}, \ldots, \mathbf{u}_n\}$ and stack them into $\mathbf{U} \in \mathbb{R}^{H \times S}$, which serves as input to the temporal architecture to capture temporal dependencies and generate predictions for future packet delays. We use two commonly used deep-learning frameworks for this task: (i) a recurrent architecture (e.g. LSTM) and (ii) an attention-based architecture (e.g. Transformer). The output of the temporal architecture is

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{n+l}\}_{l=0}^{L-1} \in \mathbb{R}^{L \times V}, \tag{3.14}$$

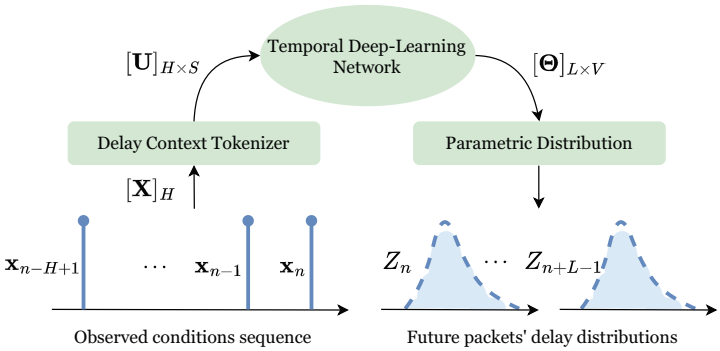


Figure 3.3: Temporal-based learning and prediction approach overview

where each $\boldsymbol{\theta}_{n+l}$ is a vector of size V and fully characterizes the distribution for the delay of packet $n+l$.

For training this network we construct a dataset in the form

$$\mathcal{D} \;=\; \left\{(\mathbf{x}_{m-H+1:m}, z_{m:m+L-1})\right\}_{m=H}^{N_D+H}, \tag{3.15}$$

where N_D is the total number of samples. This is generated by sliding a window (indexed by m) over a long, prerecorded series of observed network conditions and packet delays. The loss function is then computed based on the model's outputs θ_{m+l} for $l = 0, \ldots, L-1$ as follows:

$$\mathcal{L}(\mathcal{D}) \;=\; -\sum_{m=1}^{N_D}\sum_{l=0}^{L-1} \ln\Big(p_Z\big(z_{m+l} \mid \theta_{m+l}\big)\Big), \tag{3.16}$$

which completes the description of the temporal prediction approach.

The next sections provide detailed overviews of the included papers and explain how each one tackles the previously stated research questions. Specifically, in Section 3.2, we perform a predictability analysis on multi-hop Geo/Geo/1 queuing systems, while accounting for potential defects in the observation process. Next, in Section 3.3, we study DVP prediction in multihop queueing systems where the observable states are the queue lengths at each hop. We first explore how accurate tail predictions are obtained in delay prediction, and then investigate how to implement an AQM scheme within such queueing systems using these predictors. Finally, in Section 3.4, we extend our investigation to an actual 5G network, first focusing on modeling and collecting delay-related data and then applying our delay prediction techniques to the operational 5G setup.

3.2 Delay Predictability in Queuing Systems (Paper A)

In this paper, after introducing our definition of predictability, we apply the framework to a system whose conditions follow a first-order Markov dependency, where each subsystem's state has a stationary relationship with its conditional delay distribution, providing a suitable basis for analyzing end-to-end delay prediction.

Our study explores the impact of key parameters, such as the number of hops, the randomness of state transitions, and various observational defects, on the system's predictability. We further extend this analysis to Geo/Geo/1/K queues, modeling each subsystem's queue length as part of the observed state and examining how these Markovian queues influence our predictability metrics.

Moreover, predictability is treated as a function of the lead time: by increasing the lead time, we examine how far into the future the forecast distribution—conditioned on observing the Markov chain's current state—deviates from the marginal (prior) distribution.

Key contributions of this paper are:

- Introducing a formal definition of predictability and quantifying it using total variation distance.
- Deriving exact and approximate expressions for predictability in Markov-modulated multi-hop communication systems such as Geo/Geo/1/K queues.
- Developing spectral-based upper bounds for predictability, providing insights into the relationship between system dynamics and predictability.
- Evaluating the trade-offs between predictability and system observability, highlighting practical implications for observability in communication networks.

Our findings aim to inform the design of next-generation communication systems, emphasizing the importance of predictability as a metric for proactive adaptation and reliable performance.

The conceptualization of this paper was done by the author of this thesis in collaboration with Prof. James Gross. Analytical proofs and the implementation of the simulations were carried out by the author of this thesis. The analysis of the resulting data and writing the paper was carried out in collaboration with the co-authors.

3.2.1 System Model

We formulate a delay predictability analysis using the predictability definition, in a system that operates under the Markovian assumption—meaning that the current state X_n is sufficient to predict future states such as X_{n+1}. This eliminates the reliance on extensive historical data which simplifies the analytical model and $X_n^{(m)}$ will form a finite, discrete-time Markov chain with transition probabilities

captured by a transition matrix P. Moreover, we assume a stationary stochastic dependency between each subsystem's state and its performance:

$$\forall L, \Pr(Z_{n+L} \mid X_{n+L} = x) = \Pr(Z_n \mid X_n = x), \tag{3.17}$$

for all possible states $x \in \mathcal{X}$. This means the conditional probability distribution of performance given the state remains constant over time.

An example is when each subsystem is modeled as a Markovian queue such as Geo/Geo/1/K. We analyze the predictability for these systems where the observable condition is the queue length. In these queues, the arrival process follows a Bernoulli distribution with arrival probability α, and the service process is geometrically distributed with service probability μ. The first goal is to derive analytical expressions or bounds for predictability that account for Markovian transitions and end-to-end delay in such systems.

Regarding the observations of the system state, they might be delayed, so that $O_n = X_{n-d}$ for some delay d, or they may be aggregated via a surjective mapping $\phi : X \to A$ into a coarser state space A. This observation model is crucial for analyzing defects in the monitoring system, as every delay prediction relies on the quality and timeliness of the measured data. The second goal is to analyze how delay predictability is affected by these defects using analytical expressions or approximations.

3.2.2 Analysis

We aim to derive exact, approximate, and upper bounds for predictability using the proposed framework in Markovian conditions. The analysis spans single-hop and multi-hop systems, exploring the effects of imperfect observations and system configurations. We show in the paper that for systems governed by Markovian dynamics, the forecast distribution is derived as:

$$\Pr(Z_{n+L} \mid X_n = x) = \sum_{y \in \mathcal{X}} P^L(x, y) \Pr(Z_{n+L} \mid X_{n+L} = y), \tag{3.18}$$

where $P^L(x, y)$ is the L-step state transition probability of the Markov chain from state x to y. Furthermore, the marginal distribution of the performance metric is derived by:

$$\Pr(Z_{n+L}) = \sum_{y \in \mathcal{X}} \pi(y) \Pr(Z_{n+L} \mid X_{n+L} = y), \tag{3.19}$$

where $\pi(y)$ is the stationary distribution of the Markov chain. Using these derivations and predictability definition in Equation 3.8, the predictability can be derived exactly using:

$$D_n(L) = \frac{1}{2} \sum_{z \in \mathcal{Z}} \left| \sum_{y \in \mathcal{X}} \big(P^L(x, y) - \pi(y)\big) \Pr(Z_{n+L} \mid X_{n+L} = y) \right|. \tag{3.20}$$

To bound the predictability, we use spectral properties of reversible Markov chains, leveraging the relationship between eigenvalues and the system's convergence behavior. In Theorem 2 of the paper, we show that predictability of delay in such systems is bounded as:

$$D_n(L) \leq \frac{1}{2}\left(\sum_{j=2}^{|\mathcal{X}|} \lambda_j^{2L} f_j^2(x)\right)^{1/2} \sqrt{2(R-1)}, \tag{3.21}$$

where:

- λ_j are the eigenvalues of the Markov chain's transition matrix P, sorted with descending order, with $\lambda_1 = 1$,
- $f_j(x)$ are the eigenfunctions corresponding to λ_j,
- $R = \frac{\sum_z \sum_y \pi(y) r_y^2(z)}{\sum_y \pi(y) r_y(z)}$, with $r_y(z) \triangleq \Pr(Z_n = z \mid X_n = y)$.

A more simplified version of this bound is achieved when using the second-largest eigenvalue $\lambda^* \triangleq \lambda_2$ as:

$$D_n(L) \leq \frac{1}{2}\lambda_*^L \left(\frac{1}{\pi(x)} - 1\right)^{1/2} \sqrt{2(R-1)}. \tag{3.22}$$

This form emphasizes the geometric decay of predictability over time, driven by the spectral gap $\xi = 1 - \lambda_2$.

When considering multi-hop scenarios, the total variation distance exhibits a subadditivity property, implying that the system's overall predictability is bounded by the sum of each hop's predictability. Formally,

$$D_n^{\text{multi-hop}}(L) \leq \sum_{m=1}^{M} D_n^{(m)}(L), \tag{3.23}$$

where $D_n^{(m)}(L)$ represents the predictability of individual hops. This result highlights how predictability in multi-hop systems depends on the predictability of each hop and their observability.

Furthermore, we summarize our analysis on scenarios with imperfect observations as follows:

- Delayed Observations: When observations lag by d time slots ($O_n = X_{n-d}$), the predictability becomes equivalent to that of a system with perfect observations but with a lead time $L + d$.

- Aggregated Observations: State aggregation reduces the Markov chain's state space, creating a new transition matrix $\bar{P}$:

$$\bar{P}(a,b) = \sum_{x \in \phi^{-1}(a)} \sum_{y \in \phi^{-1}(b)} \frac{\pi(x)P(x,y)}{\pi(\phi^{-1}(a))}. \tag{3.24}$$

  Predictability under aggregation reflects the loss of information due to the reduced state space.

- Partial Observations: For a subset of observable states, unobserved states contribute zero to the total variation distance, reducing overall predictability.

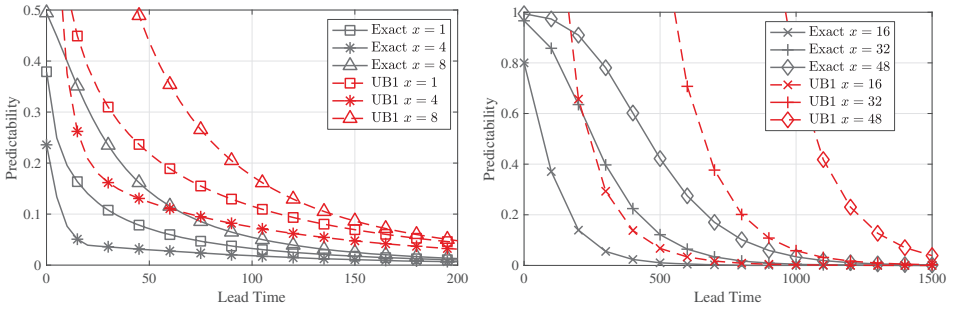
Moreover, we apply the exact predictability framework to Geo/Geo/1/K queues and derive a closed-form approximation that reveals how queue parameters—such as arrival and service rates—affect predictability. We evaluate this approximation in conjunction with upper bounds and exact solutions for single-hop and multihop Geo/Geo/1/K systems under a range of parameter settings. Lastly, we examine predictability in multihop queues across different observation scenarios.

3.2.3 Numerical Results

In order to highlight the main findings of our analysis of Markov-modulated queues and delay predictability, we now present several key numerical experiments. These experiments validate our exact derivations, approximation methods, and upper bounds, and also illustrate how partial observability and bottlenecks can shape overall predictive performance. All predictability curves in the following figures show a decay with increasing lead time, reflecting the general principle that predictive power diminishes as we look further into the future. Throughout these figures, we consider a single-hop Geo/Geo/1/K queue with default parameters, including an arrival rate of $\alpha = 0.32$, a service rate of $\mu = 0.4$, and a capacity $K = 128$, unless otherwise stated.

First, we examine Figure 3.4, which contrasts the exact predictability curves from Theorem 1 (solid lines) with the spectral-based upper bound from Theorem 2 (dashed lines). For moderate queue lengths, the bound aligns reasonably well with the exact results. At higher queue lengths, however, it grows looser because it reflects a broad range of transitions, many of which are less relevant when the queue persists in large states.

Additionally, Figure 3.5 explores a three-hop tandem Geo/Geo/1/K setup featuring a bottleneck queue. Four different observation configurations are shown: observing all queues, observing only the first queue, observing only the last queue, or observing only the bottleneck queue. Monitoring just the bottleneck queue preserves most of the end-to-end predictability, because the total delay is largely dominated by the most congested queue. Observing a non-bottleneck queue alone yields less predictive utility.



(a) (b)

Figure 3.4: Predictability (solid lines) versus the spectral-based upper bound (dashed lines) from Theorem 2, for different observed states in a single-hop Geo/Geo/1/K queue.

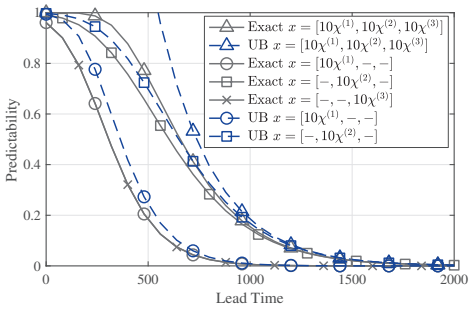


Figure 3.5: Predictability of a three-hop tandem Geo/Geo/1/K queue with a bottleneck in the middle (service rates of 0.4, 0.38, and 0.4, and arrival rate 0.34). Four observation scenarios are shown: all queues, only the first queue, only the last queue, and only the bottleneck queue.

Overall, these results show that exact derivations, approximation methods, and spectral-based bounds each have a useful role depending on the system state and level of detail required. The upper bound provides a general but sometimes loose envelope for the system's predictive potential. In multi-hop scenarios, knowing only the bottleneck queue state can recover most of the benefits of full observability, revealing practical guidelines for system monitoring and resource allocation.

3.3 Delay Prediction and Control in Queuing Systems

In this section, we analyze probabilistic delay prediction and control within queuing systems. We model each subsystem as a single-server queue with an infinite buffer operating under a first-in, first-out (FIFO) discipline. Packets arrive at the end node and traverse these queues sequentially. For each packet n in queue m, we denote the waiting time as $W_n^{(m)}$, the service time as $S_n^{(m)}$, and define the sojourn time as

$$Z_n^{(m)} = W_n^{(m)} + S_n^{(m)}. \tag{3.25}$$

We denote the stationary distribution of the service time in queue m by $F_{S_m}(s)$; however, the detailed characteristics and higher-order moments of these distributions remain unknown. Each packet is also associated with a target delay τ_n, representing the maximum allowable time from its arrival until it must be processed to meet the application's delay requirements.

In the delay prediction task, our goal is to estimate the DVP for packet n at any time t during its journey through the network. This probability is defined as the likelihood that the packet's total delay exceeds its target τ_n, conditioned on the current network state. Formally, we define:

$$\phi_{n,t} = \Pr\big(Z_n > \tau_n \mid Z_n \geq t - T_n,\ X_t = \mathbf{x}_t\big), \tag{3.26}$$

where X_t represents the network state at time t (for instance, the queue lengths) and $\mathbf{x}_t$ is its observed realization. The objective is to accurately estimate this conditional probability.

In this context, the delay prediction task represents a transient form of posterior distribution estimation as in Equation 3.5. Unlike the standard posterior prediction—which conditions solely on the current or historical observations—this transient prediction additionally accounts for the elapsed time since the packet's generation. Consequently, it dynamically adjusts the delay violation probability as the packet traverses the network, thereby capturing the evolving delay behavior in real time.

For the delay control task, we focus on a single queue at time t containing C packets. The performance metric is defined as the fraction of packets that are processed within their target delay:

$$R_t = \frac{1}{C} \sum_{i=1}^{C} \mathfrak{I}\big[Z_i \leq \tau_i\big], \tag{3.27}$$

where $\mathfrak{I}[\cdot]$ is the indicator function, equal to 1 if a packet's sojourn time does not exceed its target delay and 0 otherwise.

In this system, servers can drop packets from their queues—a mechanism known as AQM—if processing them is predicted to result in a delay violation. For each packet i in the queue, a binary decision variable $A_{i,t}$ is introduced; $A_{i,t} = 0$ indicates that the packet is dropped, while $A_{i,t} = 1$ means it is admitted for service. Packets

that are not dropped are processed according to a general stationary service time distribution $F_S(s)$, which captures the inherent randomness of the wireless channel conditions.

The central challenge in delay control is to design an AQM policy that maps the current state of the queue—specifically, the remaining delay budgets—to an optimal packet-dropping decision vector $A_t = \{A_{i,t}\}_{i=1}^{C}$. Mathematically, the problem is formulated as:

$$a_t^* = \arg\max_a \mathbb{E}\Big[R_t \mid A_t = a\Big], \tag{3.28}$$

where a is a candidate decision vector specifying the dropping decisions for all packets in the queue.

3.3.1 Transient DVP Prediction in Queuing Systems (Paper B)

In existing literature, techniques for calculating the DVP of packets often rely on network calculus [62], which imposes strong assumptions on channel models and independence of service times across hops—conditions that are difficult to maintain in practical wireless networks. Consequently, data-driven methods have gained traction for predicting the delay distribution; however, most focus primarily on the bulk of the delay distribution rather than the critical tail probabilities.

In this work, we present a transient DVP prediction framework based on MDNs, enhanced by an EVT-based tail model (i.e., a generalized Pareto distribution) to improve tail accuracy. Under general stationary service processes—including those with heavy-tailed service times—our approach accurately captures both the main body and the tail of the delay distribution, surpassing conventional GMMs. Moreover, it operates without imposing restrictive assumptions about arrival processes.

The contributions of this paper are summarized as follows:

- We propose a data-driven transient DVP predictor that seamlessly integrates EVT-based tail modeling with MDN-based density estimation.
- We implement and validate the predictor in a multi-hop queuing network to estimate DVP based on instantaneous network state variables.
- We demonstrate that our approach surpasses state-of-the-art methods, especially in scenarios with limited training data and for stringent reliability requirements (e.g., guarantee levels above 10^{-2}).

The author of this thesis developed and implemented the EVM prediction method. The queuing model was developed in collaboration with the co-authors. The author of this thesis carried out the implementation of the simulations and machine learning (ML) pipeline. The analysis of the resulting data and paper writing was carried out in collaboration with the co-authors.

Approach

Our goal is to estimate the transient DVP defined in (3.26). To achieve this, we simplify the full transient delay distribution—which is conditioned on the time of observation—by converting it into a conditional distribution of the remaining delay Z^{REM} given the network state ahead of the packet, denoted as X^{REM}. Specifically, we approximate

$$\Pr\big(Z_n = z \mid Z_n \geq t - T_n,\, X_t = \mathbf{x}_t\big) \approx \Pr\big(Z^{\mathrm{REM}} = z - (t - T_n) \mid X^{\mathrm{REM}} = \mathbf{x}_t^{\mathrm{REM}}\big), \quad (3.29)$$

where $\mathbf{x}_t^{\mathrm{REM}}$ is derived by subtracting the number of packets following the target packet from the full state $\mathbf{x}_t$.

To estimate this conditional distribution, we use the non-temporal delay prediction approach with EVM introduced earlier. Therefore, the transient DVP is approximated as:

$$\phi_{n,t} \approx p_Z\big(z > \delta_{n,t};\, \theta = h_\omega(\mathbf{x}_n)\big), \quad (3.30)$$

where h_ω denotes the fully connected neural network, $p_Z(z \mid \theta)$ denotes the parametric density with parameters θ, and the remaining delay budget $\delta_{n,t}$ for packet n at time t is computed as:

$$\delta_{n,t} = \max\Big(\tau_n - (t - T_n),\, 0\Big). \quad (3.31)$$

Numerical Results

In this subsection, we study the performance of our proposed EVM-based predictor for transient DVP estimation in a three-hop tandem queueing network. We compare it against baseline GMM-based predictors, including an extended GMM version that increases the number of Gaussian components.

We implement a simulation in MATLAB (Simulink) to collect data on the queue lengths and delays for each hop in a 3-hop tandem queue with periodic arrivals. The system operates at an arrival rate of $\alpha = 0.9$, with all three servers having identical service rates $\mu = 1$. Service times follow a heavy-tailed distribution spliced from a Gamma and a GPD at the 0.8 quantile, where $\theta_{\mathrm{Gamma}} = 0.2$, $k_{\mathrm{Gamma}} = 5$, and $\xi_{\mathrm{GPD}} = 0.2$. Each predictor is trained in TensorFlow to estimate the conditional DVP for a range of target delays τ (corresponding to $1 - 10^{-2}$ quantile down to $1 - 10^{-5}$ quantile) under various network states.

To evaluate tail accuracy, we compute the logarithmic error,

$$\text{tail error} = \Big|\log(\Pr[Z_n > \tau \mid X_n = \mathbf{x}_n]) \;-\; \log\big(p_Z(z > \tau \mid \theta = h_\omega(\mathbf{x}_n))\big)\Big|,$$

and average this measure over different states $\mathbf{x}_n$. Below, we summarize two key comparisons using Figure 3.6 and Figure 3.7.

Figure 3.6 shows the logarithmic error of the predicted DVP for different training sample sizes. Even with fewer training samples, EVM-based predictors consistently

achieve lower error than the GMM-based counterparts, indicating better robustness under limited data. In the lower-probability region, the EVM-based approach remains close to the true tail behavior even down to 10^{-4}, whereas GMM methods begin to diverge significantly below 10^{-2}.

In Figure 3.7, we compare GMM predictors with increasing numbers of Gaussian components to assess whether adding more complexity narrows the gap in tail predictions. Although adding components helps reduce the GMM's overall error, the EVM-based predictor continues to outperform these refined GMM variants, particularly in the extreme tail region. Moreover, increasing the number of Gaussian components demands more training data and leads to higher computational cost, yet still does not match EVM accuracy.

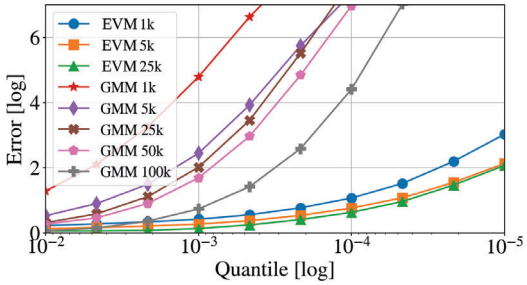


Figure 3.6: Logarithmic error of DVP prediction for different training sample sizes. EVM-based predictors maintain lower errors than GMM-based predictors, especially in the tail region.

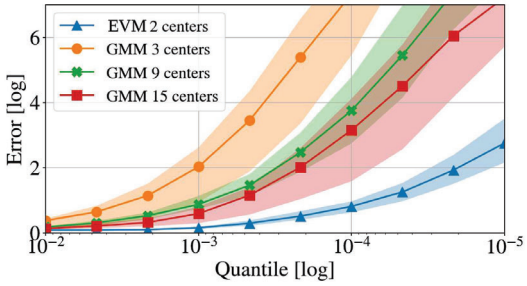


Figure 3.7: Comparison of GMM predictors with different numbers of Gaussian components. Despite increased complexity, GMM predictors still lag behind the EVM-based predictor in tail prediction. Performance variations shown for 8 different trained models per curve.

The main observations from these results are as follows. First, the EVM-based predictor achieves significantly higher tail accuracy compared to GMM-based methods, especially when estimating probabilities below 10^{-2}. Second, the EVM-based

approach is robust to limited sample sizes, preserving its advantage when data are scarce. Third, increasing the number of Gaussian components does reduce GMM errors but does not bridge the gap in performance, while also making training more computationally demanding. These findings affirm that the proposed EVM-based method is well suited for capturing rare delay events in latency-critical applications, where reliable tail estimation is crucial for proactive network management.

3.3.2 Delay Control with DVP Predictors in Queuing Systems (Paper C)

In this work, we leverage transient DVP predictions to manage queue delay by selectively dropping packets according to their remaining delay budget and the status of the queue. This form of decision-making is addressed via AQM algorithms. However, many existing AQM schemes rely on static delay thresholds and complex parameter tuning, rendering them less effective for dynamic and stochastic wireless links. To address these limitations, in this paper we proposed *Delta*, a novel AQM framework that integrates real-time DVP predictions into packet dropping decisions. Rather than relying on fixed queues or thresholds, Delta dynamically adapts to current network conditions, reducing delay violations without necessitating frequent parameter re-tuning. By drawing on probabilistic predictions, Delta is particularly well-suited to wireless links with inherently stochastic service times.

The main contributions of this paper are as follows:

- *Delta* AQM scheme, a novel algorithm that exploits real-time DVP predictions for packet dropping decisions, enabling continuous adaptation to highly dynamic network conditions.
- An extensive simulation study demonstrating that Delta outperforms established AQM methods (e.g., CoDel and DeepQ) and remains robust against variations in training sample size and potential mismatches between predicted and actual service processes.

The author of this thesis developed the AQM algorithm in collaboration with Prof. James Gross and Prof. György Dán. The implementation of the simulations and ML-pipeline was carried out by the author of this thesis and Neel Roy. The analysis of the resulting data and writing the paper was carried out in collaboration with all co-authors.

Approach

We begin by simplifying the optimization function in (3.28). The expected performance—i.e., the fraction of packets that meet their target delay—conditioned on a given dropping decision vector $A_t = a$ is expressed as

$$\mathbb{E}\left[R_t \mid A_t = a\right] = \frac{1}{C}\sum_{i=1}^{C}\mathbb{E}\left[\mathfrak{I}\left(Z_i \leq \tau_i\right) \mid A_t = a\right] = \frac{1}{C}\sum_{i=1}^{C}\psi_{i,t,a}, \tag{3.32}$$

where $\psi_{i,t,a}$ denotes the transient success probability for packet i under the dropping vector a.

This transient success probability is approximated by converting the conditional delay distribution into one of the remaining delay. In particular, we write

$$\psi_{i,t,a} \approx \Pr\Big(Z^{\mathrm{REM}} < \delta_{i,t} \mid X^{\mathrm{REM}} = \sum_{j=1}^{i} a_j\Big), \tag{3.33}$$

where $\delta_{i,t}$ is the remaining delay budget for packet i at time t, and $\sum_{j=1}^{i} a_j$ represents the effective number of packets ahead of packet i after applying the dropping decisions. This conditional distribution is then estimated using the DVP predictor introduced earlier, such that

$$\psi_{i,t,a} = 1 - \varphi_{i,t,a}, \quad \text{with} \quad \varphi_{i,t,a} \approx p_Z\Big(z > \delta_{i,t};\, \theta = h_\omega\Big(\sum_{j=1}^{i} a_j\Big)\Big). \tag{3.34}$$

The overall objective function to be maximized is defined as

$$\Psi_a = \sum_{i=1}^{C} \psi_{i,t,a}, \tag{3.35}$$

and the optimal dropping vector a^* is chosen as the one that maximizes Ψ_a. Essentially, this amounts to a search over all possible combinations of dropping decisions for the packets in the queue (i.e., evaluating every possible binary vector of decisions as shown in Figure 3.8). Although this algorithm has an exponential time complexity of $O(2^N)$, it can be implemented on a limited subset of the queue (e.g., the first 15 packets).

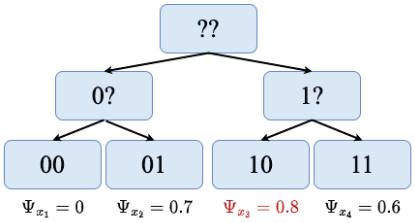


Figure 3.8: Decision tree for dropping packets in a queue of length 2 (0 indicates drop, 1 indicates pass).

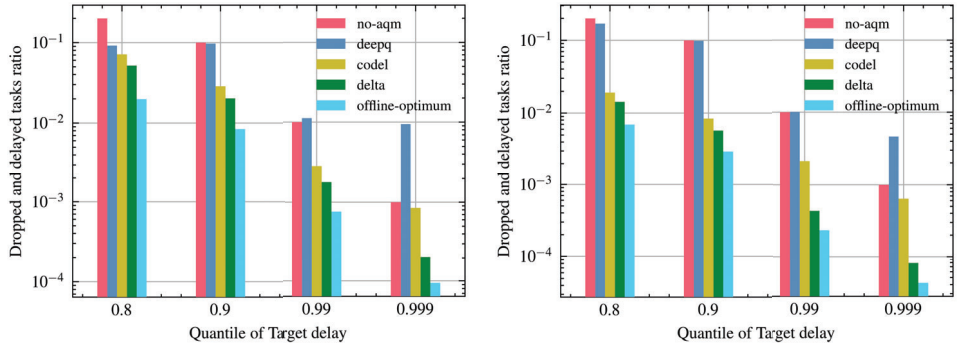
Numerical Results

The Delta AQM scheme is evaluated in a simulated single-hop queuing environment and its performance is benchmarked against established methods such as

CoDel [84] and DeepQ [85], as well as an offline-optimum policy. The evaluation focuses on how well each scheme meets the target delay under varying network utilizations.

Figure 3.9 shows the performance comparison using two subfigures. In subfigure 3.9a, the system operates at a utilization factor of 91.6%, while subfigure 3.9b corresponds to a higher utilization of 96.7%. In both scenarios, the metric of interest is the fraction of packets failing to meet their target delay. The results indicate that Delta consistently achieves a lower failure ratio compared to both CoDel and DeepQ. This superior performance is attributed to Delta's adaptive, data-driven approach which efficiently adjusts the dropping decisions in real time.

In greater detail, under moderate network loads (91.6% utilization), Delta reduces delay violations significantly over traditional AQM methods. As network load increases to 96.7%, the benefits of Delta become even more pronounced. The adaptive mechanism of Delta—relying on real-time DVP predictions—allows it to preemptively drop packets that are unlikely to meet the delay targets, thereby reducing congestion and improving overall performance. This advantage is particularly critical in wireless networks where the variability in service times can otherwise lead to severe performance degradation if not properly managed.



(a) Utilization: 91.6%

(b) Utilization: 96.7%

Figure 3.9: Performance comparison of Delta, CoDel, DeepQ, and the offline-optimum policy.

The evaluation results clearly demonstrate that Delta's parameter-free, adaptive approach not only simplifies deployment in wireless networks but also results in a substantial reduction in delay violations. This is achieved without the need for the extensive manual tuning that is characteristic of conventional AQM schemes. In summary, Delta provides a robust solution that is well-suited for the stochastic and highly dynamic nature of modern wireless systems.

3.4 Delay Analysis and Prediction in 5G

In this section, we extend our investigation to real 5G networks, focusing on the delay in the 5G network and leveraging data-driven methods to both predict delay and provide insights for delay optimization. To adhere to the system-of-systems perspective on end-to-end delay—as illustrated in Figure 3.1—we decompose the end-to-end delay Z_n of packet n traversing a 5G network into the sum of three major components:

$$Z_n = Z_n^{(\mathrm{C})} + Z_n^{(\mathrm{Q})} + Z_n^{(\mathrm{L})}, \tag{3.36}$$

where each component captures distinct delay sources along the packet's journey.

Core Delay ($Z_n^{(\mathrm{C})}$) represents the delay incurred in the core network, which becomes particularly significant when the network gateway is several hops away from the RAN. The conditions that exclusively influence this component are denoted by $X_n^{(\mathrm{C})}$, which may include factors such as the number of hops, total bandwidth usage in the core, and overall congestion levels.

Queuing Delay ($Z_n^{(\mathrm{Q})}$) arises from buffering in the radio link control (RLC) queue. The associated conditions, represented by $X_n^{(\mathrm{Q})}$, are critical to capture—most notably, the queue length at the moment the packet enters the system.

Link Delay ($Z_n^{(\mathrm{L})}$) is the delay experienced on the radio link, which we further decompose into three subcomponents:

$$Z_n^{(\mathrm{L})} = Z_n^{(\mathrm{Lt})} + Z_n^{(\mathrm{Ls})} + Z_n^{(\mathrm{Lr})}. \tag{3.37}$$

Here, $Z_n^{(\mathrm{Lt})}$ denotes the transmission delay (i.e., the time required to transmit a block regardless of the decoding result), $Z_n^{(\mathrm{Ls})}$ represents the segmentation delay caused by dividing a packet into smaller segments when the transmission block size is smaller than the packet size, and $Z_n^{(\mathrm{Lr})}$ corresponds to the retransmission delay incurred when decoding failures—due to poor channel conditions—trigger hybrid automatic repeat request (HARQ) retransmissions. The conditions influencing the link delay, denoted by $X_n^{(\mathrm{L})}$, include channel-related metrics such as the modulation and coding scheme (MCS) index and signal to interference and noise ratio (SINR), as well as scheduling-related factors like available bandwidth.

The challenges addressed in this section are twofold. First, we focus on the measurement and decomposition of end-to-end delay in a real 5G system according to the above model, which enables us to quantify the contribution of each delay component to overall delay violations. Second, we apply our delay prediction techniques—incorporating both previously developed methods for abstract models and novel approaches tailored specifically for 5G—to predict delay in an operational 5G environment.

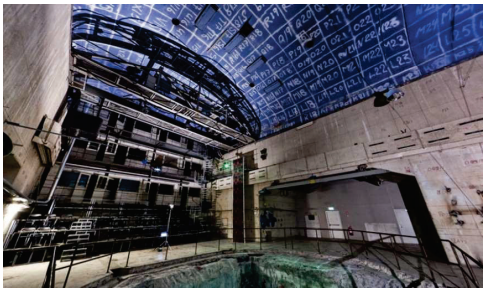


Figure 3.10: R1 hall, an underground site at KTH Royal Institute of Technology where experimental platform for edge computing applications (ExPECA) testbed is installed.

3.4.1 Delay Measurement and Analysis in 5G (Papers D and E)

A key requirement for data-driven delay modeling and prediction is an experimental platform that can execute time-sensitive applications on a flexible 5G stack while providing access to all delay-relevant signals throughout the protocol layers. We tackle this requirement in Paper D by implemnting ExPECA, a fully controllable 5G-enabled edge-computing testbed that enables reproducible wireless end-to-end experimentation. Our main contributions in Paper D are:

- **End-to-end, cloud-native 5G testbed.** ExPECA fuses software-defined radio (SDR) + commercial off-the-shelf (COTS) radios, containerised edge/cloud compute, and Openstack-based orchestration to run time-sensitive workloads on a fully customizable computation and communication stack.

- **Reproducible wireless experiments due to isolated location.** ExPECA sits 25 m below ground in KTH's R1 hall, shielding experiments from external interference and giving researchers a stable RF environment for reproducible latency studies.

In Paper D, the author of this thesis together with Manuel Olguín Muñoz and Prof. James Gross designed the testbed and its capabilities. Then the installation and implementation of the testbed components was carried out by the author of the thesis in collaboration with all co-authors.

Building on this platform and motivated by Research Question 4, in Paper E we introduce the end-to-end delay analytics framework (EDAF) framework, which its purpose is to efficiently collect relevant raw trace data and convert them into actionable latency insights. Its key contributions are:

- **Fine-grained delay-decomposition toolkit.** EDAF inserts timestamp hooks from the application layer down to HARQ, funnels them to a cen-

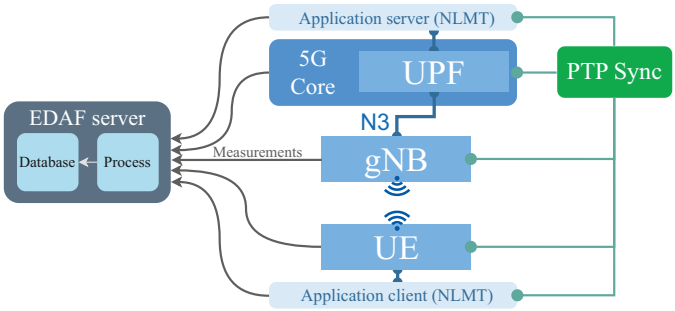


Figure 3.11: EDAF implementation setup

tral server, and reconstructs per-packet core, queuing, and link-delay components—delivering instant, actionable latency breakdowns.

- **Open-source, real-time analytics implementation.** Implemented as an open-source micro-service package on OpenAirInterface 5G, EDAF delivers real-time plots of delay complementary cumulative censity functions (CCDFs) and component shares, instantly surfacing latency bottlenecks and enabling on-the-fly network tuning.
- **Proved impact on an OAI-5G uplink.** Evaluation of EDAF on ExPECA's SDR 5G setup let us spot and remove segmentation and frame-alignment delays, cutting average end-to-end latency from 12 ms to 4 ms.

In Paper E, the author of the thesis designed and implemented the framework on OpenAirInterface 5G uplink stack in collaboration with Marius Tillner. The experiments were carried out by the author of the thesis. The analysis of the resulting data and writing the paper was carried out with the help of Gourav Prateek Sharma and Prof.James Gross.

EDAF Design and Implementation

Leveraging the SDR-based OpenAirInterface 5G network in the ExPECA testbed, we deployed EDAF and carried out uplink experiments to gauge its performance and usefulness, with probabilistic delay analysis as the principal objective. Specifically, EDAF operates as follows:

1. **Instrumentation and Data Collection.** Within the OpenAirInterface stack, timestamps are captured whenever a packet enters or exits a major processing step:
    - RLC Queue arrival and departure (queuing delay)
    - Scheduling process (scheduling delay),

- HARQ processes (link-level retransmission delays),
- Application layer arrival and departure (end-to-end (E2E) delays).

Each timestamp is merged with a packet identifier and extensive network-state parameters (e.g., MCS index, allocated physical resource blocks (PRBs), queue depth), then forwarded to the EDAF server for aggregation and storage in a database (as illustrated in Figure 3.11).

2. **Analysis and Decomposition.** Once the data is stored in a time-series repository, EDAF correlates the timestamps across all nodes and protocol layers, reconstructing the journey of every packet. This enables EDAF to decompose end-to-end delays into component-level contributions—as described in the system model, core delay, queuing delay, and link delay compnents—and link them to specific radio settings or scheduling policies.

   We identify the degree that each component contributes to delay violations for a given delay target τ via

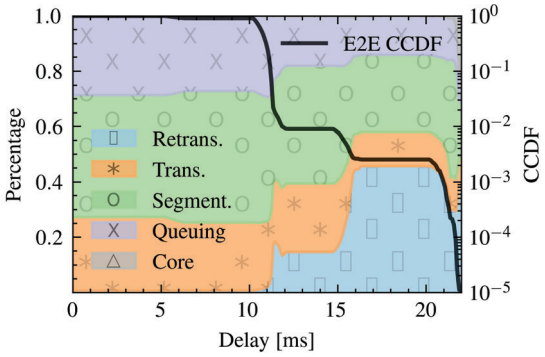
$$\mathbb{E}\left[\frac{Z_n^{(m)}}{Z_n} \mid Z_n > \tau, X_n\right]. \tag{3.38}$$

3. **Near Real-Time Insights.** By making these detailed metrics available almost immediately, EDAF supports both offline and online data processing. In an offline context, researchers can systematically benchmark various network configurations. In an online context, EDAF could be integrated with 5G control-plane functions (e.g., network data analytics function (NWDAF)) that dynamically adapt scheduling or resource allocations based on measured latency and reliability.

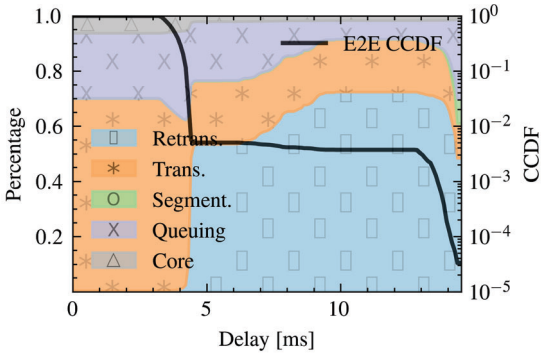
Through this combined approach, ExPECA's reproducible experimentation environment and EDAF's fine-grained instrumentation address both the *observability* and *controllability* requirements that are essential for reliable latency studies. Researchers can isolate and manipulate individual factors—like channel bandwidth, packet arrival offsets, or scheduling intervals—and then measure the resulting effects on per-packet delay distributions.

Analysis and Results

In order to evaluate the effectiveness of EDAF in delay measurement and decomposition, we conducted a series of E2E latency measurements on a 5G uplink scenario powered by SDR-based OpenAirInterface. The system was configured with a fixed MCS index of 23 to ensure consistent modulation and coding rates, and packets of size 500 bytes were sent at regular intervals to represent moderate data traffic between two SDR nodes at ExPECA. We also tested different uplink PRBs allocations and varied packet arrival times relative to the time division duplex



(a) Default configuration



(b) Optimized configuration

Figure 3.12: EDAF E2E CCDF and decomposition in experiments feature a fixed MCS index of 23 and 500-byte packets. Optimized configuration eliminates segmentation delay by adding 5 PRBs and optimized packet arrival times

(TDD) schedule. These adjustments are representative of real-world tuning practices, such as configuring radio resource blocks to match user demand and aligning traffic injections with favorable transmission slots to minimize queuing. Throughout the experiments, EDAF recorded per-packet timestamps across the RLC and medium access control (MAC) layers, as well as additional trace information (e.g., queue size, HARQ attempts), enabling a detailed decomposition of the E2E delay.

Figure 3.12 presents representative outcomes of EDAF's analysis under two key configurations. Subfigure 3.12a shows the default system settings, where insufficient PRBs force each packet to be segmented. As the CCDF curve indicates, a significant fraction of overall latency stems from queueing and segmentation overhead, with HARQ retransmissions accounting for a noticeable portion of the distribution's tail. In addition, the bars (or stacked areas) within the figure highlight how EDAF apportions the total delay into multiple components: queueing at the RLC buffer,

transmission and retransmission within the MAC/PHY layers, segmentation time, and any additional core-network delay. By contrast, Subfigure 3.12b illustrates the effects of increasing the uplink allocation by five PRBs (eliminating segmentation) and shifting packet arrivals to align more closely with the TDD uplink windows. Under these optimized settings, segmentation delay nearly disappears, and queueing time is reduced, resulting in a leftward shift of the CCDF curve and a marked decrease in packets surpassing tight delay targets such as 5 ms and 15 ms.

These results underscore how segmentation often dominates average values of latency, while occasional HARQ retransmissions lead to heavier tails, especially in higher-load or poorer-channel conditions. By isolating each component's contribution, EDAF helps identify which factors—multi-segment transmission, frame alignment, or resource allocation—are most detrimental to meeting a given delay target. In the illustrated example, simply increasing the number of PRBs and adjusting traffic arrival timing reduced the proportion of late packets, demonstrating that data-driven insight into E2E delay behavior can effectively guide targeted network optimizations in 5G systems.

In summary, these experiments underscore how a combination of reproducible testbed deployments (ExPECA) and per-packet analytics (EDAF) can significantly advance our ability to detect, understand, and mitigate the key causes of E2E delay in time-sensitive 5G networks.

3.4.2 Delay Prediction in 5G (Papers F and G)

As outlined in our general system model, delay prediction can be approached at both non-temporal (prior or posterior) and temporal (forecast) levels. In this section, we extend our machine learning-based delay prediction methods to real 5G systems, evaluating their performance on two distinct platforms: (i) a COTS 5G system, and (ii) an SDR-based OpenAirInterface (OAI) deployment at the ExPECA testbed. Main contributions in these papers are:

- **Data-Driven Tail Analysis:** In our earlier work implemented on queueing systems (Paper B), we integrated EVT and MDNs to model the high-latency tail in a more precise and flexible manner. In Paper F, we analyze this approach in real 5G systems on end-to-end packet delays and show that it successfully captures extreme latency events (as rare as 10^{-5}), without relying on specific channel or traffic analytical models.
- **Multi-Horizon Prediction:** In Paper G, we introduce our temporal prediction methods, which is capable of estimating the delay distribution for several upcoming packets simultaneously, from the historical network condition observations using LSTMs and Transformers. This novel approach offers proactive insights for resource allocation and scheduling in time-sensitive scenarios.
- **Context Encoding and Temporal Modeling:** In Paper G, we propose a systematic tokenization approach to represent both instantaneous and historical network conditions in a fixed-dimensional format, enabling LSTM or Transformer-based temporal models to learn the short and long-term dependencies of packet delays in a more efficient manner.
- **Real-World Implementation & Efficiency Study:** We implement and benchmark all predictor variants of both papers on two 5G platforms—a COTS system and an SDR-based OAI testbed—using the ExPECA and EDAF frameworks. Alongside demonstrating accuracy gains in realistic conditions, we also examine model scalability, training time, and data requirements to assess overall learning efficiency.

The author of the thesis designed and implemented the prediction algorithms in both papers. In Paper F the experiments were designed and carried out by the author of the thesis in colaboration with Gourav Prateek Sharma and Prof. James Gross while in Paper G the experiments were carried out in collaboration with Ahmad Traboulsi and Gourav Prateek Sharma. Writing the papers was done in collaboration with all co-authors in each paper.

Table 3.1 lists the prediction targets and methods addressed in each study. Paper F focuses on how accurately our models capture the tail of the delay distribution for both prior and posterior cases on a COTS 5G platform. For the posterior case, we test whether having the MCS index of the 5G link as a conditioning variable can

reveal different tail behaviors and how well the MDN-based predictors can capture them.

Paper G turns to temporal forecasting: using the SDR-based 5G network with EDAF, we predict the future delay distribution from an expanded feature set: packet size, inter-arrival time, arrival slot, MCS index, and counts of HARQ and RLC retransmissions. Forecast quality is then judged by empirical NLL and mean absolute error (MAE) on the test datasets.

Beyond prediction accuracy, we also evaluate several efficiency metrics. First, we assess *data efficiency* by measuring how much training data is needed to reach a given accuracy level. Second, we track *training time* under consistent hardware conditions to gauge each model's scalability. Finally, we observe the growth in *model size*—specifically, the number of parameters—and how accuracy scales, since excessive parameter growth can hinder real-world deployment. All metrics are evaluated on dedicated test datasets to ensure an unbiased measure of generalization.

Table 3.1
Summary of Delay Prediction Studies on Real 5G Systems in this Thesis

Delay Dist.	**Predictor**	**System**	**Paper**
Prior $\Pr(Z_n)$	Tail-Focused EVM	COTS 5G	F
Posterior $\Pr(Z_n \mid X_n)$	Tail-Focused MLP EVM	SDR 5G	F
Forecast $\Pr(Z_{n+L} \mid X_{n-H+1:n})$	Temporal LSTM/Transformer GMM	SDR 5G	G

The following sections provide a detailed description of these approaches and present our experimental results.

Tail-Focused Results (Paper F)

In Paper F, the evaluation centers on data gathered from two 5G platforms: a COTS-based private 5G system and an SDR-based OAI 5G deployment. In both cases, millions of latency samples were collected; in the COTS measurements we changed the user equipment (UE)'s position, whereas in the OAI tests we swept through different MCS levels. The predictive models—GMM-based MDNs and the proposed EVM—were trained on a subset of these samples and then tested against unseen data to assess tail accuracy. Key performance indicators include how closely each model's predicted latency distribution aligns with the empirical distribution at extreme quantiles (e.g., above the 99.999% level).

Figure 3.13 compares the empirical CCDF of uplink latencies from the COTS 5G network to the fitted tail probability curves from both a GMM-based MDN

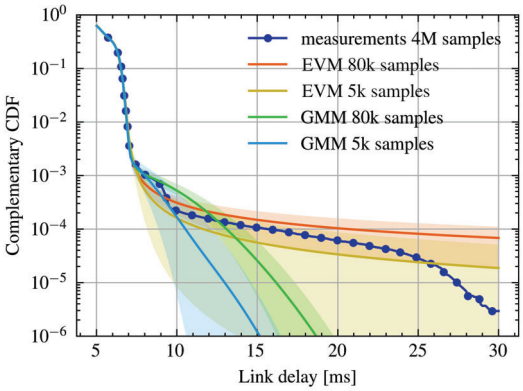


Figure 3.13: COTS 5G uplink latency measurements vs parametric density fits with different number of samples and models

and the EVM approach. Each point on the CCDF indicates the fraction of packets whose delay exceeds a given threshold, thereby illustrating the tail behavior. The figure shows that, although a purely Gaussian mixture provides a reasonable approximation up to moderate quantiles, it diverges substantially at rarer events (e.g., probabilities on the order of 10^{-4} or 10^{-5}). By contrast, the EVM model closely tracks the empirical tail, accurately capturing the slow decay of extreme delays. This improvement remains evident so long as the training set is large enough—roughly 10^4 samples or more—to expose the model to tail behavior.

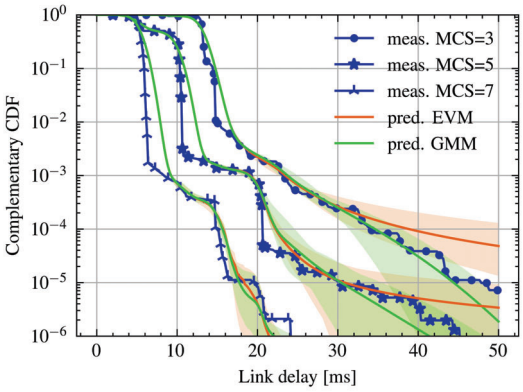


Figure 3.14: SDR 5G uplink latency tail probability measurements conditioned for different MCS indices with 5M samples vs predictions of MDN models trained with 250k samples (5%)

The evaluation also investigates two further aspects. First, different sample sizes are used to train the models (e.g., 10^3, 10^4, 10^5), allowing the study of how data

scarcity affects each estimator's reliability in the high-latency region. EVM generally demonstrates higher robustness at smaller sample sizes, though performance still degrades when sample counts are very low. Second, the OAI 5G measurements incorporate multiple MCS settings, which serve as a condition variable in the MDN. Figure 3.14 show that EVM again excels at reproducing observed latency distributions, especially once the more "bumpy" empirical tails are smoothed by noise regularization. These results highlight the importance of both accurate tail modeling and careful data preprocessing when predicting extreme latencies in wireless networks.

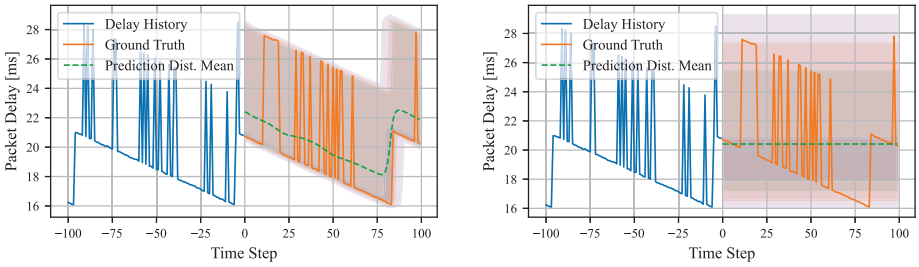
Numerical Results on Temporal Prediction (Paper G)

In this section, we provide an overview of how our temporal machine learning models (namely LSTM and Transformer) perform when used for delay prediction in a 5G SDR environment at ExPECA testbed, as originally introduced in Paper G.

In the evaluations, we systematically vary the historical window size, the forecast horizon length, and the amount of training data to investigate how these key parameters affect model accuracy in practice. In particular, we emphasize the amount of data needed to achieve a specific accuracy target for a given prediction horizon. Moreover, we include delay traces produced from experiments with different packet interval times in the datasets. Two baseline approaches are included for reference: (i) a MLP that observes only the most recent condition vector and outputs the delay distribution for the entire upcoming set of L packets, and (ii) an LSTM-SS model that leverages the entire history but still produces a single distribution covering the next L packets collectively. These baselines reflect standard delay-prediction strategies found in current literature. Our main contributions are the per-packet LSTM and Transformer predictors, which produce an individual delay distribution for each future packet in the horizon.

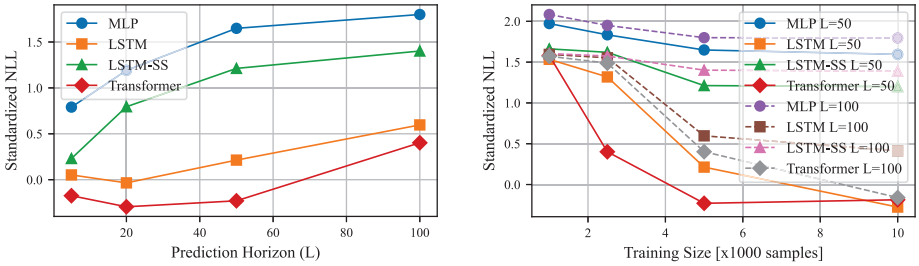
First, we illustrate the benefit of multi-step delay forecasting with the example in Figure 3.15. It shows 200 consecutive time steps—100 historical and 100 future—where the packet delay follows a sawtooth pattern due to TDD misalignment and also exhibits occasional retransmission spikes of about 7.5 ms. Subfigure 3.15a demonstrates that the multi-step Transformer adapts its predicted distributions at each future time step, indicating, for instance, a near-zero probability of exceeding 24 ms around step 75. By contrast, the single-step MLP (Subfigure 3.15b) treats all future time steps uniformly, offering less nuanced forecasts. This highlights how multi-step probabilistic methods can better capture evolving conditions and enable more proactive responses.

Figures 3.16a compares model accuracy at different forecast horizons. The Transformer achieves the lowest NLL, effectively capturing long-range dependencies. Multi-step LSTM ranks second but sees its NLL climb at larger horizons ($L = 100$). Meanwhile, single-step methods (LSTM-SS, MLP) lag in accuracy, though LSTM-SS benefits from historical data and improves over simpler feedforward baselines. Figure 3.16b investigates how training set size affects NLL. Longer



(a) Multi-step transformer predictor (b) Single-step MLP predictor

Figure 3.15: Comparison of multi-step transformer and single-step MLP predictors on a test sample with 100 historical and 100 future time steps, trained on a 10k dataset. The multi-step model produces time-varying distributions (colored regions for 50%, 70%, 90%, and 99% coverage, from dark to light) that closely cover the ground truth, while the single-step model yields uniform forecasts.



(a) Prediction accuracy (NLL) against horizon (5k dataset).

(b) Prediction accuracy (NLL) against dataset size.

Figure 3.16: Comparison of model accuracy across different prediction horizons and training dataset sizes.

horizons ($L = 100$) predictably lead to higher uncertainty, but all models improve with more data. In particular, both multi-step LSTM and Transformer gain significantly when moving from 1k to 10k samples, underscoring the importance of sufficient training data for reliable distributional forecasts. These results highlight the robust performance of multi-step distributional forecasting in dynamic 5G scenarios, with the Transformer delivering consistently better accuracy and calibration when sufficient training data is available.

CHAPTER 4

Conclusions & Future Work

4.1 Conclusions

This dissertation contributes to the literature on latency prediction for upcoming wireless networks such as 6G. We claimed that forecasting the full delay distribution is the appropriate strategy due to its flexibility and structured our work around this idea to guide future wireless-system design. To summarise, we revisit the four objectives set at the outset: Delay Predictability (O1), Delay Prediction (O2), Delay Measurement & Data Collection (O3), and Delay Control (O4). We adopted a two-fold methodology to approach these objectives. First, abstract queuing-theoretic models let us test our new tools in a controlled setting, revealing key factors and fundamental insights into delay predictability and prediction. Second, a practical 5G testbed allowed us to pursue the objectives in a real network context. Linking the objectives with these theoretical and experimental setups produced research questions that we addressed through separate studies. The remainder of this conclusion outlines those questions, our answers across the studies, and their implications.

For Research Question 1, which examined how to define predictability in communication networks, we conducted a study that produced a new, network-specific definition. The definition gauges a prediction system's predictability by evaluating the informational utility of its observations. By analyzing first-order Markov models—including single-hop and multi-hop Geo/Geo/1/K queues—we derived closed-form formulas and spectral upper bounds for predictability. These results show how hop count, transition randomness, and observation imperfections (e.g., delay or aggregation) shape predictability. The resulting framework clarifies the fundamental limits of latency forecasting and supports the design of more predictable networks. A typical implementation challenge is selecting appropriate network fea-

tures as predictors. The predictability framework lets us rank candidate parameter sets by comparing their forecast distributions with the corresponding marginals. It also identifies operating states where predictability breaks down, enabling us to deactivate prediction resources when they are no longer useful.

Building upon this foundation, we developed a suite of data-driven techniques for probabilistic delay prediction to address Research Questions 2 and 5 (delay prediction in queues and 5G respectively). A key innovation is our tail-optimized prediction method, which integrates EVT within a MDN framework. This approach significantly enhances the accuracy of predicting rare, high-latency events – a critical requirement for ultra-reliable low-latency communication (URLLC) scenarios. We first evaluated the method on multi-hop queueing systems, then on end-to-end packet delays in a real 5G network. Results show that our tail-oriented approach achieves higher accuracy with less data. Extensive measurements on the COTS 5G setup revealed a heavy-tailed delay distribution, which our method modelled more precisely. The two-stage process also indicated that noise regularization and data normalization are key to accurate fits in the queuing environment, where the controlled setting made troubleshooting and tuning straightforward. We further advanced the state-of-the-art by developing temporal prediction models, leveraging both LSTM and Transformer architectures, to capture the time-varying dynamics of wireless networks and provide multi-step forecasts. Our tokenization approach enabled scalable and robust encoding of network context, improving the efficiency and generalizability of our models.

To address Research Question 3 (AQM) and show the practical value of probabilistic delay forecasts for proactive network control, we developed Delta, a new AQM that feeds real-time DVP estimates into its drop logic. Delta adjusts on the fly by comparing each queued packet's DVP with its delay budget, and our tests in queuing simulations show it cuts delay violations far more than established AQM schemes such as CoDel. Crucially, it achieves this without heavy parameter tuning, making it well-suited to wireless links. A key element in 5G networks is the queuing mechanism and its QoS-driven management which is still an open research area to which this study adds a new algorithm.

To tackle Research Questions 4 and 5 (delay analytics and prediction in 5G respectively) we created ExPECA, an underground 5G-edge testbed that pairs OpenStack resource reservation with Kubernetes-orchestrated containers, shielded from external interference and equipped with SDRs. This setup yielded the high-resolution latency traces for EDAF framework developed for Research Question 4 and enabled end-to-end delay measurement and network observability to support latency prediction in Research Question 5. Because its radios, channels and workloads are fully programmable, ExPECA is now a reusable platform for studying deterministic wireless, TSN integration and data-driven latency prediction under repeatable conditions. In effect, it converts our theoretical contributions into a practical, reproducible tool for forthcoming low-latency 6G research.

On the other hand, the EDAF framework was developed to address Research Question 4 specifically and provides a foundation for precise, real-time analysis of

end-to-end delay in 5G networks. By enabling fine-grained delay decomposition, real-time visualization, and integration within a live SDR-based OpenAirInterface 5G system, EDAF allows researchers and operators to pinpoint latency bottlenecks and a causal analysis. These capabilities not only facilitate systematic delay analysis but also open the door to data-driven delay control and optimization strategies in time-sensitive wireless applications.

In conclusion, this dissertation has advanced the state-of-the-art in latency prediction and control, providing both theoretical foundations and practical solutions for the next generation of cellular networks. In the next section, we examine the thesis's broader impacts, including its societal and economic aspects.

4.2 Broader Impact

The advancements in latency prediction and control for 5G and beyond networks, as presented in this thesis, have far-reaching implications that extend beyond technical improvements. By enabling ultra-reliable low-latency communication (URLLC), the proposed data-driven approaches support critical applications such as industrial automation and extended reality (XR), which are increasingly integral to modern society. These developments contribute to societal progress, economic growth, and environmental sustainability, as outlined below.

Societally, the research on latency prediction and control in 5G and beyond networks, enhances the accessibility and safety of critical applications; for instance, extended reality (XR) technologies can provide immersive learning experiences, improving educational engagement, while smart city innovations like autonomous vehicles and intelligent traffic systems enhance urban safety and reduce congestion. Economically, it optimizes industries reliant on real-time data processing, such as manufacturing and logistics, by improving efficiency, reducing downtime, and enabling better resource allocation, fostering productivity and driving innovation through integration with edge computing and Internet of Things (IoT) ecosystems. From a sustainability perspective, the data-driven latency management techniques contribute to more energy-efficient wireless networks by enabling smarter resource allocation and reducing over-provisioning, while supporting applications like smart grids and intelligent traffic systems that promote reduced urban emissions and improved energy efficiency.

Future work could explore several promising avenues which we elaborate in the following section.

4.3 Future Work

This research has established a solid foundation for analyzing network performance and prediction accuracy. Moving forward, several promising avenues have emerged to bridge theoretical frameworks with practical network implementations. Our proposed future work aims to refine prediction models, optimize observability,

and advance delay and queue management techniques in 5G and multihop networks. These efforts are expected to not only enhance overall system performance but also provide a deeper understanding of the underlying network dynamics.

1. **Using Predictability Framework:** The growing demand for QoS and quality-of-experience (QoE) prediction in wireless networks creates new opportunities to examine the predictability framework more deeply. Future researchers could apply the framework to determine which observability features yield predictive gain for a particular metric—such as delay or throughput. In addition, because monitoring systems often employ time aggregation to reduce overhead, investigating how this reduced observability affects predictability remains an important topic.

2. **Implementing Delta AQM on 5G Network:** Future research on AQM can expand our Delta AQM scheme to real 5G links. Delta can replace the conventional AQM schemes on the RLC or packet data convergence protocol (PDCP) queues to make dropping decisions and improve delay charachteristics. Additionally, dynamic assignment of packets to different quality-of-service queues can be investigated in Delta to further enhance service differentiation. Testing these methods on real networks—with their inherent service stochasticity and time-sensitive tasks—will provide critical insights into practical performance.

3. **Improving Tail-Focused Delay Prediction:** For EVM approach, further investigation into advanced techniques is needed to improve tail fitting reliability. Some preliminary experiments have shown high errors in the tail region with EVM approach. However, initial results suggest that employing a multi-step tail fitting strategy—where bulk training is decoupled from tail fitting—could address performance discrepancies. This approach warrants further exploration with more sophisticated methods.

4. **Refining Delay Measurement and Modeling in 5G:** We have already enhanced the EDAF framework to extract fine-grained timestamps, capturing events such as scheduling handshakes, RLC acknowledgments, and resource allocation decisions. Future work can focus on performing causal analyses on this multidimensional data to automatically uncover underlying dependencies. Causal analysis, in this context, means identifying true cause-and-effect relationships between these events, going beyond simple correlations. For example, it would help us determine if a slow RLC acknowledgement directly causes increased delay, or if both are influenced by a separate underlying factor. Such insights, gained by understand the why behind performance issues, are expected to lead to more confident and accurate delay predictions, because prediction models will take into account the true derviers of network delay.

5. **Optimizing Temporal Delay Prediction:** Regarding temporal delay prediction, our coverage plots demonstrated that both the Transformer and LSTM approaches exhibited a tendency towards over-coverage. While this is beneficial in scenarios where prediction reliability is paramount, it can lead to resource over-provisioning if these predictions are used for allocation purposes. Therefore, future work should explore more sophisticated methods to mitigate this issue. Some existing research focuses on coverage-based loss functions, often incorporated alongside the NLL loss. A promising future research direction is to investigate the integration of the coverage-adjusted loss functions to improve the precision, or "tightness," of the coverage plots and, consequently, enhance resource allocation efficiency.

In summary, the proposed future work outlines a strategic roadmap to push the boundaries of current network performance analysis and prediction methods. Through advanced modeling techniques, real-world testing, and innovative analytical approaches, we aim to bridge the gap between theoretical frameworks and practical implementations. These developments are expected to enhance prediction reliability, optimize resource management, and provide valuable insights into the dynamics of modern wireless networks.

References

[1] Maria A Lema, Andres Laya, Toktam Mahmoodi, Maria Cuevas, Joachim Sachs, Jan Markendahl, and Mischa Dohler, "Business case and technology analysis for 5G low latency applications", *IEEE Access*, vol. 5, pp. 5917–5935, 2017.

[2] Mohammed S Elbamby, Cristina Perfecto, Chen-Feng Liu, Jihong Park, Sumudu Samarakoon, Xianfu Chen, and Mehdi Bennis, "Wireless edge computing with latency and reliability guarantees", *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1717–1737, 2019.

[3] Hongyu Pei Breivold and Kristian Sandström, "Internet of things for industrial automation – challenges and technical solutions", in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, 2015, pp. 532–539.

[4] Junaid Ansari, Christian Andersson, Peter de Bruin, János Farkas, Leefke Grosjean, Joachim Sachs, Johan Torsner, Balázs Varga, Davit Harutyunyan, Niels König, *et al.*, "Performance of 5G trials for industrial automation", *Electronics*, vol. 11, no. 3, p. 412, 2022.

[5] GrandViewResearch, "Edge Computing Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Services, Edge-managed Platforms), By Application, By Industry Vertical, By Region, And Segment Forecasts, 2023 - 2030", Tech. Rep. GVR-2-68038-106-1, 2022, p. 254.

[6] ETSI, "5G; NR; NG-RAN; architecture description", ETSI, Tech. Rep. ETSI TS 123 501 V16.8.0, 2021.

[7] "D1.1: DETERMINISTIC6G Use cases and Architecture Principles", DETERMINISTIC6G, DETERMINISTIC6G Project Deliverable, Jun. 2023, [Online] Available: https://deterministic6g.eu/index.php/library-m/deliverables/.

[8] 5GAA Automotive Association *et al.*, "Making 5G proactive and predictive for the automotive industry", *White Paper, Aug*, 2019.

[9] ETSI, "Multi-access edge computing (MEC); phase 2: Use cases and requirements", ETSI, Tech. Rep. ETSI GS MEC 002 V2.1.1, 2018.

[10] "D2.1: First report on 6G-centric Enablers", DETERMINISTIC6G, DETERMINISTIC6G Project Deliverable, Dec. 2023, [Online] Available: https://deterministic6g.eu/index.php/library-m/deliverables/.

[11] "D3.1: Report on 6G Convergence Enablers Towards Deterministic Communication Standards", DETERMINISTIC6G, DETERMINISTIC6G Project Deliverable, Dec. 2023, [Online] Available: https://deterministic6g.eu/index.php/library-m/deliverables/.

[12] 3GPP, "Medium Access Control (MAC) protocol specification", 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.321, May 2019, Version 15.5.0.

[13] Hussein Al-Zubaidy, Jörg Liebeherr, and Almut Burchard, "Network-layer performance analysis of multihop fading channels", *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 204–217, 2016.

[14] Ahmed Nasrallah, Akhilesh S Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2018.

[15] U Narayan Bhat, *An introduction to queueing theory: modeling and analysis in applications*. Springer, 2008, vol. 36.

[16] Donald Gross, John F Shortle, James M Thompson, and Carl M Harris, *Fundamentals of queueing theory*. John wiley & sons, 2011, vol. 627.

[17] David G Kendall, "Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain", *The Annals of Mathematical Statistics*, pp. 338–354, 1953.

[18] S Asmussen, *Applied probability and queues*, 2003.

[19] Jim Gettys, "Bufferbloat: Dark buffers in the internet", *IEEE Internet Computing*, vol. 15, no. 3, pp. 96–96, 2011.

[20] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[21] Richelle Adams, "Active queue management: A survey", *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1425–1476, 2012.

[22] Arkaitz Bitorika, Mathieu Robin, Meriel Huggard, and Ciaran Mc Goldrick, "A comparative study of active queue management schemes", in *Proc. of ICC*, 2004.

[23] Kathleen Nichols and Van Jacobson, "Controlling queue delay", *Communications of the ACM*, vol. 55, no. 7, pp. 42–50, 2012.

[24] Jeffrey G. Andrews, Stefano Buzzi, Wan Choi, Stephen V. Hanly, Angel Lozano, Anthony C. K. Soong, and Jianzhong Charlie Zhang, "What will 5G be?", *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[25] Afif Osseiran *et al.*, "Scenarios for 5G mobile and wireless communications: The vision of the METIS project", *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.

[26] *NR; Overall Description; Stage-2 (3GPP TS 38.300 version 15.8.0 Release 15)*, Release 15, 3GPP, 2020.

[27] Andrea Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[28] Stefania Sesia, Issam Toufik, and Matthew Baker, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[29] Erik Dahlman, Stefan Parkvall, and Johan Skold, *5G NR: The next generation wireless access technology*. Academic Press, 2020.

[30] *5G; NR; Physical layer procedures for data (3GPP TS 38.214 version 16.2.0 Release 16)*, Release 16, 3GPP, 2020.

[31] Christopher M Bishop and Nasser M Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.

[32] Stuart Coles, Joanna Bawa, Lesley Trenner, and Pat Dorazio, *An introduction to statistical modeling of extreme values*. Springer, 2001, vol. 208.

[33] James Pickands III, "Statistical inference using extreme order statistics", *the Annals of Statistics*, pp. 119–131, 1975.

[34] Diederik P. Kingma and Jimmy Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: `1412.6980 [cs.LG]`.

[35] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15, Lille, France: JMLR.org, 2015, pp. 448–456.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[38] Jeffrey L Elman, "Finding structure in time", *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[39] S Hochreiter, "Long short-term memory", *Neural Computation MIT-Press*, 1997.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, *et al.*, "Language models are unsupervised multitask learners", *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[42] A Vaswani, "Attention is all you need", *Advances in Neural Information Processing Systems*, 2017.

[43] Kian-Ping Lim, Weiwei Luo, and Jae H. Kim, "Are US stock index returns predictable? Evidence from automatic autocorrelation-based tests", en, *Applied Economics*, vol. 45, no. 8, pp. 953–962, Mar. 2013.

[44] Christoph Bandt and Bernd Pompe, "Permutation Entropy: A Natural Complexity Measure for Time Series", *Phys. Rev. Lett.*, vol. 88, no. 17, p. 174 102, Apr. 2002, Publisher: American Physical Society.

[45] Joshua Garland, Ryan James, and Elizabeth Bradley, "Model-free quantification of time-series predictability", en, *Physical Review E*, vol. 90, no. 5, p. 052 910, Nov. 2014.

[46] Frank Pennekamp, Alison C Iles, Joshua Garland, Georgina Brennan, Ulrich Brose, Ursula Gaedke, Ute Jacob, Pavel Kratina, Blake Matthews, Stephan Munch, *et al.*, "The intrinsic predictability of ecological time series and its potential to guide forecasting", *Ecological Monographs*, vol. 89, no. 2, 2019, Publisher: Wiley Online Library.

[47] Samuel V. Scarpino and Giovanni Petri, "On the predictability of infectious disease outbreaks", en, *Nature Communications*, vol. 10, no. 1, p. 898, Feb. 2019.

[48] Andrés Abeliuk, Zhishen Huang, Emilio Ferrara, and Kristina Lerman, "Predictability Limit of Partially Observed Systems", *Scientific Reports*, vol. 10, no. 1, Nov. 2020, Publisher: Nature Publishing Group.

[49] Yong Li, Depeng Jin, Pan Hui, Zhaocheng Wang, and Sheng Chen, "Limits of Predictability for Large-Scale Urban Vehicular Mobility", *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2671–2682, Dec. 2014.

[50] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási, "Limits of predictability in human mobility", *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010. eprint: https://www.science.org/doi/pdf/10.1126/science.1177170.

[51] Taro Takaguchi, Mitsuhiro Nakamura, Nobuo Sato, Kazuo Yano, and Naoki Masuda, "Predictability of conversation partners", English, *Physical Review X*, vol. 1, no. 1, pp. 1–16, 2011.

[52] William Bialek, Ilya Nemenman, and Naftali Tishby, "Predictability, Complexity, and Learning", *Neural Computation*, vol. 13, no. 11, pp. 2409–2463, Nov. 2001.

[53] Kyle Haven, Andrew Majda, and Rafail Abramov, "Quantifying predictability through information theory: Small sample estimation in a non-gaussian framework", *Journal of Computational Physics*, vol. 206, no. 1, pp. 334–362, 2005.

[54] Guopeng Li, Victor L. Knoop, and Hans van Lint, "Estimate the limit of predictability in short-term traffic forecasting: An entropy-based approach", *Transportation Research Part C: Emerging Technologies*, vol. 138, p. 103 607, 2022.

[55] Yu-Hsueh Fang and Chia-Yen Lee, *Predictability analysis of regression problems via conditional entropy estimations*, 2024. arXiv: 2406.03824 [cs.LG].

[56] Timothy DelSole, "Predictability and Information Theory. Part I: Measures of Predictability", en, *Journal of the Atmospheric Sciences*, vol. 61, no. 20, pp. 2425–2440, Oct. 2004.

[57] V. Krishnamurthy, "Predictability of Weather and Climate", *Earth and Space Science*, vol. 6, no. 7, pp. 1043–1056, Jul. 2019.

[58] Timothy DelSole and Michael K. Tippett, "Predictability: Recent insights from information theory", en, *Reviews of Geophysics*, vol. 45, no. 4, Dec. 2007.

[59] Guoru Ding, Jinlong Wang, Qihui Wu, Yu-dong Yao, Rongpeng Li, Honggang Zhang, and Yulong Zou, "On the Limits of Predictability in Real-World Radio Spectrum State Dynamics: From Entropy Theory to 5G Spectrum Sharing", *IEEE Communications Magazine*, vol. 53, no. 7, pp. 178–183, Jul. 2015, Conference Name: IEEE Communications Magazine.

[60] Sihai Zhang, Junyao Guo, Tian Lan, Rui Sun, and Jinkang Zhu, "Real entropy can also predict daily voice traffic for wireless network users", in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[61] Fei Jing, Chuang Liu, Jian-Liang Wu, and Zi-Ke Zhang, "Toward structural controllability and predictability in directed networks", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 12, pp. 7692–7701, 2022.

[62] Jaya Prakash Champati, Hussein Al-Zubaidy, and James Gross, "Transient analysis for multihop wireless networks under static routing", *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 722–735, 2020.

[63] Baldomero Coll-Perales, M. Carmen Lucas-Estañ, Takayuki Shimizu, Javier Gozalvez, Takamasa Higuchi, Sergei Avedisov, Onur Altintas, and Miguel Sepulcre, "End-to-End V2X Latency Modeling and Analysis in 5G Networks", *IEEE Transactions on Vehicular Technology*, vol. 72, no. 4, pp. 5094–5109, 2023.

[64] Rouba Ibrahim and Ward Whitt, "Real-time delay estimation based on delay history", *Manufacturing & Service Operations Management*, vol. 11, no. 3, pp. 397–415, Jul. 2009.

[65] Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum, "Queue mining – predicting delays in service processes", in *Advanced Information Systems Engineering*, Cham: Springer International Publishing, 2014, pp. 42–57.

[66] Majid Raeis, Ali Tizghadam, and Alberto Leon-Garcia, "Predicting distributions of waiting times in customer service systems using mixture density networks", *International Conference on Network and Service Management (CNSM)*, pp. 1–6, 2019.

[67] Darlan C. Moreira, Igor M. Guerreiro, Wanlu Sun, Charles C. Cavalcante, and Diego A. Sousa, "QoS Predictability in V2X Communication with Machine Learning", in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, May 2020, pp. 1–5.

[68] Sokratis Barmpounakis, Lina Magoula, Nikolaos Koursioumpas, Ramin Khalili, Jose Mauricio Perdomo, and Ramya Panthangi Manjunath, "LSTM-based QoS prediction for 5G-enabled Connected and Automated Mobility applications", in *2021 IEEE 4th 5G World Forum (5GWF)*, Oct. 2021, pp. 436–440.

[69] Xiaozheng Dang, Di He, and Cong Xie, "A Time Delay Prediction Model of 5G Users Based on the BiLSTM Neural Network Optimized by APSO-SD", *Journal of Electrical and Computer Engineering*, vol. 2023, B. Rajanarayan Prusty, Ed., pp. 1–20, Jun. 2023.

[70] Ali Safari Khatouni, Francesca Soro, and Danilo Giordano, "A Machine Learning Application for Latency Prediction in Operational 4G Networks", in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Apr. 2019, pp. 71–74.

[71] Daniel Fabian Külzer, Firas Debbichi, Sławomir Stańczak, and Mladen Botsov, "On Latency Prediction with Deep Learning and Passive Probing at High Mobility", in *ICC 2021 - IEEE International Conference on Communications*, Jun. 2021, pp. 1–7.

[72] Akhila Rao *et al.*, "Prediction and exposure of delays from a base station perspective in 5G and beyond networks", in *Proceedings of the ACM SIGCOMM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases*, ser. 5G-MeMU '22, New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 8–14.

[73] Akhila Rao, Hassam Riaz, Aleksandr Zavodovski, Rami Mochaourab, Viktor Berggren, and Andreas Johnsson, "Generalizable One-Way Delay Prediction Models for Heterogeneous UEs in 5G Networks", in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, May 2024, pp. 1–9.

[74] Alexandros Palaios *et al.*, "Machine Learning for QoS Prediction in Vehicular Communication: Challenges and Solution Approaches", *IEEE Access*, vol. 11, pp. 92 459–92 477, 2023.

[75] Wenhan Zhang, Mingjie Feng, Marwan Krunz, and Haris Volos, "Latency prediction for delay-sensitive V2X applications in mobile cloud/edge computing systems", in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[76] Shang Yimeng, Liu Jianhua, Ma Jian, Qiu Yaxing, Zhang Zhe, and Liu Chunhui, "A Prediction Method of 5G Base Station Cell Traffic Based on Improved Transformer Model", in *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Oct. 2022, pp. 40–45.

[77] Tahmina Azmin, Mohamad Ahmadinejad, and Nashid Shahriar, "Bandwidth prediction in 5G mobile networks using informer", in *2022 13th International Conference on Network of the Future (NoF)*, IEEE, 2022, pp. 1–9.

[78] Haris Volos, Takashi Bando, and Kenji Konishi, "Latency Modeling for Mobile Edge Computing Using LTE Measurements", in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug. 2018, pp. 1–5.

[79] Diyar Fadhil and Rodolfo Oliveira, "Estimation of 5G Core and RAN End-to-End Delay through Gaussian Mixture Models", *Computers*, vol. 11, no. 12, p. 184, Dec. 2022.

[80] Christofer Flinta, Wenqing Yan, and Andreas Johnsson, "Predicting round-trip time distributions in IoT systems using histogram estimators", in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Budapest, Hungary: IEEE, Apr. 2020, pp. 1–9.

[81] Forough Shahab Samani, Rolf Stadler, Christofer Flinta, and Andreas Johnsson, "Conditional Density Estimation of Service Metrics for Networked Services", *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 2350–2364, Jun. 2021.

[82] Marco Skocaj, Francesca Conserva, Nicol Sarcone Grande, Andrea Orsi, Davide Micheli, Giorgio Ghinamo, Simone Bizzarri, and Roberto Verdone, *Data-driven Predictive Latency for 5G: A Theoretical and Experimental Analysis Using Network Measurements*, Jul. 2023. eprint: 2307.02329 (cs).

[83] Rong Pan, Preethi Natarajan, Chiara Piglione, Mythili Suryanarayana Prabhu, Vijay Subramanian, Fred Baker, and Bill VerSteeg, "PIE: A lightweight control scheme to address the bufferbloat problem", in *2013 IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*, 2013, pp. 148–155.

[84] Kathleen Nichols and Van Jacobson, "Controlling queue delay", *Commun. ACM*, vol. 55, no. 7, pp. 42–50, Jul. 2012.

[85] Minsu Kim, Muhammad Jaseemuddin, and Alagan Anpalagan, "Deep Reinforcement Learning Based Active Queue Management for IoT Networks", *Journal of Network and Systems Management*, vol. 29, no. 3, p. 34, Apr. 2021.

[86] Jingling Liu, Jiawei Huang, Wenchao Jiang, Zhaoyi Li, Yijun Li, Wenjun Lyu, Wanchun Jiang, Jiao Zhang, and Jianxin Wang, "End-to-end congestion control to provide deterministic latency over internet", *IEEE Communications Letters*, vol. 26, no. 4, pp. 843–847, 2022.

[87] Sounak Kar, Bastian Alt, Heinz Koeppl, and Amr Rizk, *Optimal decision making in active queue management*, 2023. arXiv: 2202.10352 [cs.PF].

[88] Mikel Irazabal, Elena Lopez-Aguilera, Ilker Demirkol, Robert Schmidt, and Navid Nikaein, "Preventing RLC buffer sojourn delays in 5G", *IEEE Access*, vol. 9, pp. 39 466–39 488, 2021.

[89] Alexandros Stoltidis, Kostas Choumas, and Thanasis Korakis, "Active queue management in disaggregated 5G and beyond cellular networks using machine learning", in *2024 19th Wireless On-Demand Network Systems and Services Conference (WONS)*, 2024, pp. 113–120.

[90] 3GPP, "Management and orchestration; 5G performance measurements", 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.321, Jul. 2022, Version 17.7.1.

[91] 3GPP, "Management and orchestration; 5G end to end Key Performance Indicators (KPI)", 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 28.554, May 2024, Version 18.5.0.

[92] 3GPP, "5G; 5G System; Network Data Analytics Services; Stage 3", 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 29.520, Jul. 2024, Version 18.6.0.

[93] Arjun Balasingam, Manikanta Kotaru, and Paramvir Bahl, "Application-Level service assurance with 5g RAN slicing", in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, Santa Clara, CA: USENIX Association, Apr. 2024, pp. 841–857.

[94] Matthias Frei, Piotr Karbownik, Reinhard German, and Anatoli Djanatliev, "Accessing the edge: Delay evaluation to distributed edge services in a city-level 5G network", in *2024 IEEE International Conference on Cloud Engineering (IC2E)*, 2024, pp. 197–205.

[95] Panagiotis K. Gkonis, Nikolaos Nomikos, Panagiotis Trakadas, Lambros Sarakis, George Xylouris, Xavi Masip-Bruin, and Josep Martrat, "Leveraging network data analytics function and machine learning for data collection, resource optimization, security and privacy in 6G networks", *IEEE Access*, vol. 12, pp. 21 320–21 336, 2024.

[96] Khen Bo Kan, Hyunsu Mun, Guohong Cao, and Youngseok Lee, "Mobile-LLaMA: Instruction fine-tuning open-source LLM for network analysis in 5G networks", *IEEE Network*, vol. 38, no. 5, pp. 76–83, 2024.

[97] Wei Ye *et al.*, "Dissecting carrier aggregation in 5G networks: Measurement, QoE implications and prediction", in *Proceedings of the ACM SIGCOMM 2024 Conference*, ser. ACM SIGCOMM '24, Sydney, NSW, Australia: Association for Computing Machinery, 2024, pp. 340–357.

[98] Davide Scano, Francesco Paolucci, Koteswararao Kondepu, Andrea Sgambelluri, Luca Valcarenghi, and Filippo Cugini, "Extending P4 in-band telemetry to user equipment for latency- and localization-aware autonomous networking with AI forecasting", *Journal of Optical Communications and Networking*, vol. 13, no. 9, pp. D103–D114, 2021.

[99] David Larrabeiti, Luis M. Contreras, Gabriel Otero, José Alberto Hernández, and Juan P. Fernandez-Palacios, "Toward end-to-end latency management of 5g network slicing and fronthaul traffic (invited paper)", *Optical Fiber Technology*, vol. 76, p. 103 220, 2023.

[100] Haoran Wan, Xuyang Cao, Alexander Marder, and Kyle Jamieson, "Nr-scope: A practical 5g standalone telemetry tool", in *Proceedings of the 20th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '24, Los Angeles, CA, USA: Association for Computing Machinery, 2024, pp. 73–80.

[101] Flavien Ronteix–Jacquet, Alexandre Ferrieux, Isabelle Hamchaoui, Stéphane Tuffin, and Xavier Lagrange, "LatSeq: a Low-Impact internal latency measurement tool for OpenAirInterface", in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2021, pp. 1–6.

[102] Julie Carreau and Yoshua Bengio, "A hybrid pareto model for asymmetric fat-tailed data: The univariate case", *Extremes*, vol. 12, no. 1, pp. 53–76, Aug. 2008.

[103] F. F. Nascimento, D. Gamerman, and H. Lopes, "A semiparametric bayesian approach to extreme value estimation", *Statistics and Computing*, vol. 22, pp. 661–675, 2012.

Part II

Publications