Edge Computing in the Cloud: Improving Latency-Sensitive Applications with Amazon Web Services

by

MD.Sakib Hasan(20105302511)

A Research Project Progress Report submitted to the
City University
in partial fulfillment of the requirements for the degree of
Bachelor of Science in
Computer

Supervisor: Sraboni Ghosh Joya

Computer science and engineering
City University
Savar, Dhaka-1340

December 2023

DECLARATION

I hereby declare that this thesis is based on the results I have found for myself. Materials of work found by other researchers are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

MD.Sakib Hasan
ID:2015302511

CERTIFICATE

This is to certify that the thesis entitled **Edge Computing in the Cloud: Improving Latency-Sensitive Applications with Amazon Web Services** has been prepared and submitted by **MD.Sakib Hasan** in partial fulfillment of the requirement for the degree of Bachelor of Computer Science and Engineering in December 2023.

Sraboni Ghosh Joya
Supervisor

Accepted and approved in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering

ACKNOWLEDGEMENTS

I feel pleased to have the opportunity to express our heartfelt thanks and gratitude to those who rendered their cooperation in making this report.

This thesis is performed under the supervision of Sraboni Ghosh Joya, lecturer, at City University (CU), Savar, Dhaka. During the work, she has supplied us with several books, journals, and materials related to the present investigation. Without his help, kind support, and generous period she has given, I could not have performed the project work successfully in due time. First and foremost, I wish to acknowledge my profound and sincere gratitude to her for his guidance, valuable suggestions, encouragement, and cordial cooperation.

I express our utmost gratitude to the Director of CU, City University, Savar, Dhaka, for his support in completing the work within the time frame. Moreover, I would also like to thank the other faculty members of the computer science and engineering departments who have helped us directly or indirectly by providing their valuable support in completing this work.

I express our gratitude to all other sources that we have found helpful. I am indebted to those who have helped us directly or indirectly in completing this work.

Last but not least, I would like to thank all the staff of CU, City University, and my friends and family who have helped me by giving their encouragement and cooperation throughout the work.

ABSTRACT

This study investigates the transformational impact of using edge computing in the cloud, specifically focusing on improving the performance of latency-sensitive applications via Amazon Web Services (AWS), with a primary emphasis on AWS wavelength. The study addresses the growing challenges of managing edge resources, ensuring data consistency and security, and seamlessly integrating with cloud services. Our proposed approach includes a comprehensive framework for deploying and managing AWS resources such as Amazon Elastic Compute Cloud, Amazon Elastic Block Store volumes, and Virtual Private Cloud (VPC) subnets within wavelength zones, which are distinctive zones within carrier locations where wavelength infrastructure is deployed. A middleware layer orchestrates and synchronizes data processing and communication between the edge and the cloud, utilizing AWS services such as Lambda, Amazon Elastic File System (EFS), and Aurora DB within the associated AWS region. To assess the efficacy of our approach, we present a benchmark tool that measures latency, throughput, and cost across various wavelength zones worldwide. Extensive testing compares our approach to cloud-only and edge-only strategies, demonstrating significant reductions in latency, increased throughput, and cost efficiency while maintaining data security and consistency. In Bangladesh, specific recommendations are presented to improve edge applications' latency and user experience. Selecting the nearest AWS region and wavelength zone, optimizing network settings, and fostering collaboration with local Internet service providers (ISPs) and game developers are all suggestions. This collaborative effort aims to integrate AWS Wavelength, providing faster and more reliable connectivity and content delivery to end users.

Keywords: edge computing, cloud computing, Amazon Web Service, Amazon Wavelength, Internet service providers, latency-sensitive applications, AR, communication, Amazon Elastic Compute Cloud.

LIST OF ABBREVIATIONS

AWS	Amazon Web Service
AMI	Amazon Machine Images
AI	Artificial Intelligence
CNCF	Cloud Native Computing Foundation
CDK	Cloud Development Kit
CDN	Content Delivery Network
CLI	Command Line Interface
CSP	Communication Service Provider
DoS	Denial of Service
EFS	Elastic File System
EKS	Elastic Kubernetes Services
IOT	Internet of Things
AR	Augmented Reality
VR	Virtual Reality
IAM	Identity and Access Management
MAC	Multi-Access Edge computing
NAT	Network Address Translator
SSL	Secure Sockets Layer
ISP	internet service provider
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VPC	Virtual Private Cloud

LIST OF FIGURES

Figure

LIST OF TABLES

Table

TABLE OF CONTENTS

CHAPTER I

Introduction

1.1 Background of the study

In the dynamic landscape of modern computing, achieving optimal performance for latency-sensitive applications has spurred a significant focus on the amalgamation of edge computing with cloud services. This convergence, especially when orchestrated through Amazon Web Services (AWS) and the newly developed AWS Wavelength, presents engaging opportunities for addressing latency challenges in a variety of applications. As the demand for real-time responsiveness grows, this study aims to dissect and optimize the intricate interplay between edge computing and AWS, with a focus on latency-sensitive applications.

Plenty of studies have been conducted to investigate the challenges and opportunities associated with managing latency-sensitive applications in edge computing [1-8]. Liu et al.'s investigation of scheduling latency-sensitive applications in edge computing [1] and Wang et al.'s focus on optimizing latency-sensitive applications for Amazon's public cloud platform [2] are noteworthy contributions. The literature also discusses the impact of edge computing on mobile applications and how to accurately measure it [3],[4]. The work of Wang et al. on running latency-sensitive applications on public serverless edge cloud platforms adds valuable insights to this debate [5]. Research into the integration of edge computing, fog computing, and cloud computing for a variety of applications, such as smart manufacturing services, adds to the landscape [11]. The comprehensive review of computational intelligence techniques in cloud and edge computing by Asim et al. [6] provides a foundational understanding. Surveys on current research and future directions in edge computing highlight its growing importance [7].

We extend these developing research works by providing a more in-depth analysis of how AWS Wavelength helps applications that are sensitive to latency. With a particular focus on Bangladesh, we present specific recommendations tailored to the regional context, to optimize edge applications and emphasize our study's global relevance and impact beyond the empirical investigations, our research is underscored by theoretical frameworks in edge computing and cloud computing. Edge computing, characterized by decentralized computation and storage resources, seeks to minimize latency by processing data closer to the source. This is particularly crucial for applications requiring real-time responsiveness. On the other hand, cloud computing, with its centralized data centers, provides scalable and cost-effective resources but may introduce latency challenges due to geographical distances between users and servers. An overview of the typical edge computing architecture is shown in Figure 1.1.

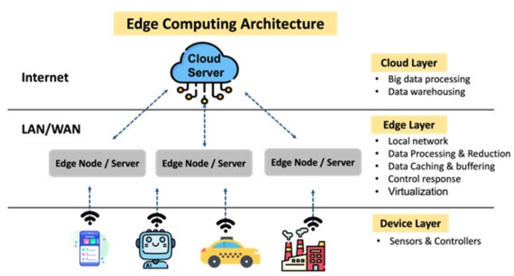

Figure 1.1: Edge Computing Architecture

.

We intend to contribute not only empirical insights but also a deeper theoretical understanding of the transformative potential of Edge Computing in the Cloud, specifically with AWS Wavelength, as we engage on this research journey. Our thorough investigation, backed up by an extensive number of references ranging from edge computing to cloud platforms, latency optimization, and regional considerations, lays the groundwork for a thorough investigation into this constantly developing field. We intend to contribute not only empirical insights but also a deeper theoretical understanding of the transformative potential of Edge Computing in the Cloud, specifically with AWS Wavelength, as we engage on this research journey. Our thorough investigation, backed up by an extensive number of references ranging from edge computing to cloud platforms, latency optimization, and regional considerations, lays the groundwork for a thorough investigation into this constantly developing field.

1.2 Objective

The following are the key objectives of this thesis report:

i AWS Wavelength is a service that allows you to deploy applications at the edge of the 5G network, reducing latency and improving performance for end users.;

ii Latency-sensitive applications are applications that require fast and responsive data processing, such as online gaming, video streaming, smart city, and e-health;

iii The Asia Pacific (Seoul) region is one of the regions where AWS Wavelength is available, and it has a Wavelength Zone that supports edge computing in the cloud;

iv The object of the study is to explore how AWS Wavelength can improve latency-sensitive applications in the Asia Pacific (Seoul) region, by using AWS services and tools, such as AWS Lambda, AWS EC2, and AWS VPC;

Overall, this thesis report aims to explore the potential of AWS Wavelength, a service that enables edge computing in the cloud, for improving latency-sensitive applications in the Asia Pacific (Seoul) region.

1.3 Research Outline

The remaining portions of the report are organized as follows:

The research unfolds across six chapters, commencing with a comprehensive introduction **Chapter I** that contextualizes the study, articulates identified issues, elucidates motivations, and outlines objectives. **Chapter II** is dedicated to an in-depth literature review, exploring 18 relevant papers and elucidating the background tools, encompassing software and hardware, employed for the research. In **Chapter III**, the study details data collection methods, feature extraction techniques, the proposed model, and the tools and environments utilized. The subsequent **Chapter IV** conducts a thorough analysis of ISP network performance, examining upload and download speeds, latency, packet loss, and jitter, with a specific focus on real-time multiplayer online gaming in Bangladesh. In **Chapter V** The research further experimentally evaluates the AWS Wavelength zone in the Asia Pacific Seoul region. Finally, **Chapter VI** synthesizes key findings, discusses expected outcomes, and outlines future research directions, offering a comprehensive conclusion and result discussion.

CHAPTER II

Literature Review

2.1 Introduction

The convergence of edge computing and cloud services stands out as a critical area of investigation in the constant realm of computing. This literature review aims to provide a summary of current research addressing edge computing integration, with a focus on Amazon Web Services (AWS) and its innovative AWS Wavelength service. Understanding the challenges, opportunities, and theoretical frameworks in this convergence is critical, given the sudden increase in demand for real-time responsiveness across diverse applications.

The review moves through studies that investigate the optimization of latency-sensitive applications, the implementation of computational intelligence, and the larger landscape of edge-cloud interplay. This preliminary research aims to inform and explain the subsequent research investigation into the transformative potential of Cloud Edge Computing, specifically through the lens of AWS wavelength

There has been much research involving this topic. Major features of this research will be discussed in this chapter.

2.2 Overview of the existing models for latency-sensitive applications

The 19 papers explore various aspects of edge computing, latency-sensitive applications, and Amazon Web Services (AWS) in the context of cloud computing. Edge computing is a paradigm that enables low-latency and high-performance applications by bringing computation and data storage closer to the sources of data. [3] introduces the concept and architecture of fog computing, which is a form of edge computing

that extends cloud computing to the edge of the network. [8] propose a smart manufacturing service system that leverages edge computing, fog computing, and cloud computing to provide real-time data analysis, decision-making, and control for various manufacturing scenarios. [9] and [10] present novel frameworks and techniques to optimize the 5G control plane for emerging edge applications, such as online gaming, augmented reality, and autonomous driving. Latency-sensitive applications are applications that require fast and deterministic response times to meet user expectations and performance service level agreements.[1],[11], and [12]propose novel scheduling algorithms and platforms to optimize the latency, cost, and quality of service of such applications in edge computing environments.
[2],[5], and [13] conduct comprehensive performance studies and evaluations of latency-sensitive applications on AWS and other public cloud platforms, such as Azure. They also propose optimization techniques and strategies to improve the efficiency and reliability of such applications on the cloud.[14] and [15] present methods to optimize cloud computations using edge computing, especially for data-intensive and stream processing applications. AWS is the world's most comprehensive and broadly adopted cloud, offering over 200 fully featured services from data centers globally. [16] presents the AWS edge computing solutions, which provide low-cost, scalable, and flexible options for delivering websites and web applications from the edge of the network. They also discuss the benefits and challenges of AWS edge computing for different use cases and scenarios.

2.2.1 Edge computing based Approaches

The advent of edge computing has ushered in a paradigm shift in the realm of computing, particularly in addressing the demands of latency-sensitive applications. This subsection provides an overview of key studies and approaches centered around edge computing, showcasing the innovative strategies employed to optimize the performance of applications requiring real-time responsiveness. In addition, features and assessment metrics employed in the various edge computing approaches are described in Table 2.1.

Reference	Selected Features	Assessment Matrices
[8]	Edge device data, Fog node data, Cloud data, Manufacturing service model	Service quality, Service cost, Service latency
[3]	Fog device data, Fog service data, Application data	Service availability, Service scalability, Service security
[9]	Edge application data, 5G control plane data, Network data	Application latency, Application throughput, Application reliability
[10]	Edge device data, Edge network data, Cellular network data	Edge service latency, Edge service consistency, Edge service availability
[1]	Edge device data, Edge application data, Network data	Application latency, Application cost, Application quality of service
[2]	Cloud application data, Cloud platform data, Network data	Application latency, Application cost, Application performance
[5]	Edge application data, Edge cloud data, Network data	Application latency, Application cost, Application scalability
[14]	Cloud application data, Edge device data, Network data	Application error, Application energy, Application latency
[11]	Edge application data, Edge device data, Network data	Application error, Application energy, Application latency
[13]	Cloud application data, Edge device data, Network data	Application error, Application energy, Application latency
[12]	Cloud application data, Edge device data, Network data	Application error, Application energy, Application latency
[15]	Edge application data, Edge cloud data, Network data	Application error, Application energy, Application latency
[17]	Cloud application data, Edge device data, Network data	Application error, Application energy, Application latency

Table 2.1: Tabular view of the edge computing approaches for low latency

Edge computing is a paradigm that enables low-latency and high-performance applications by bringing computation and data storage closer to the sources of data One of the applications that can benefit from edge computing is smart manufacturing, which involves the integration of sensors, actuators, robots, and machines to optimize the production process.[8] propose a smart manufacturing service system that leverages edge computing, fog computing, and cloud computing to provide real-time data analysis, decision-making, and control for various manufacturing scenarios. They also present a case study of a smart factory that uses their system to improve the quality and efficiency of the products. Fog computing is a term that refers to the extension of cloud computing to the edge of the network, where various devices, such as routers, gateways, and servers, can provide cloud-like services, such as computation, storage, networking, and management. [3] introduce the concept and architecture of fog computing, and discuss its potential applications, such as smart grid, smart traffic, and smart health. They also present some challenges and open issues for fog computing research and development. 5G networks are expected to enable new applications that require ultra-low latency, high bandwidth, and high reliability, such as online gaming, augmented reality, and autonomous driving. However, the current 5G control plane, which is responsible for managing the network resources and functions, may not be able to meet the diverse and dynamic demands of these applications. [9] propose a novel 5G control plane intervention framework that allows edge applications to customize and optimize the 5G control plane according to their requirements and preferences. They also demonstrate the benefits of their framework through a prototype implementation and experiments. Another challenge for edge applications is the consistency and availability of the edge-based cellular network, which may suffer from frequent handoffs, congestion, and failures. [10] present Neutrino, a fast and consistent edge-based cellular control plane that leverages a distributed consensus protocol and a network function virtualization platform to provide seamless and reliable edge services for mobile users. They also evaluate the performance and scalability of Neutrino through simulations and testbed experiments.

2.2.2 Latency-sensitive applications based approach

The cluster of papers focusing on Latency-Sensitive Applications shares a unified purpose of advancing the understanding and optimization of applications with stringent latency requirements. "Scheduling Latency-Sensitive Applications in Edge Computing" [1] explores effective scheduling strategies in edge computing environments, while "Optimizing latency-sensitive applications for Amazon's public cloud platform"[2] concentrates on tailored optimization techniques within Amazon Web Services (AWS). "Operating Latency-Sensitive Applications on Public Serverless Edge Cloud Platforms" [5] delves into the nuances of serverless edge cloud environments, and "Optimize cloud computations using edge computing" [14] explores the broader scope of cloud computations optimization through edge computing. Papers like "Towards latency-sensitive cloud-native applications: A performance study on AWS" [11] and "Cost and latency optimized edge computing platform" [13] provide comprehensive performance studies and optimization approaches on AWS. "Latency-aware strategies for deploying data stream processing applications on large cloud-edge infrastructure"[13] delves into strategies for data stream processing, while "Characterization and analysis of cloud-to-user latency: The case of Azure and AWS" [17] conducts a cross-platform analysis of cloud-to-user latency, encompassing Azure and AWS. Collectively, these papers contribute to the broader landscape of improving the performance, cost efficiency, and quality of service for latency-sensitive applications across different computing environments.

Reference	Selected Features	Assessment Matrices
[1]	Edge device data, Edge application data, Network data	Application latency, Application cost, Application quality of service
[2]	Cloud application data, Cloud platform data, Network data	Application latency, Application cost, Application performance
[5]	Edge application data, Edge cloud data, Network data	Application latency, Application cost, Application scalability
[14]	Cloud application data, Edge device data, Network data	Application error, Application energy, Application latency
[11]	Edge application data, Edge device data, Network data	Application latency, Application cost, Application quality of service
[13]	Cloud application data, Cloud platform data, Network data	Application latency, Application availability, Application scalability
[12]	Edge application data, Edge device data, Network data	Application latency, Application cost, Application quality of service
[15]	Edge application data, Edge device data, Cloud device data, Network data	Application latency, Application precision, Application recall, Application energy
[17]	Cloud application data, Cloud platform data, Network data	Cloud-to-user latency, Cloud-to-user latency distribution, Cloud-to-user latency variation

Table 2.2: Tabular view of the dictionary-based approaches for predicting results from social media

Table 2.2 describes the characteristics and evaluation criteria used by the latency-sensitive applications approaches.

[1] proposes a novel scheduling algorithm for latency-sensitive applications in edge computing, which considers the network latency, the computation latency, and the application deadline. They also present a simulation framework to evaluate their algorithm and compare it with other existing algorithms.[2] conduct a comprehensive performance study of latency-sensitive applications on Amazon's public cloud platform, and identify the key factors that affect the latency, such as the instance type, the network configuration, and the load balancer. They also propose some optimization techniques to reduce the latency and improve the performance of such applications on the cloud.[5] present a framework for operating latency-sensitive applications on public serverless edge cloud platforms, such as AWS Lambda and Azure Functions. They also discuss the challenges and opportunities of using such platforms for latency-sensitive applications and provide some guidelines and best practices for developers and operators .[14] propose a method to optimize cloud computations using edge computing, especially for data-intensive and stream processing applications. They also present a prototype system that implements their method and demonstrate its effectiveness and efficiency through experiments.[11] propose a novel edge scheduling algorithm that addresses the application latency requirements through edge computing. They also present a mathematical model and a simulation platform to analyze and evaluate their algorithm and compare it with other existing algorithms.[13] conduct a performance study of latency-sensitive cloud-native applications on AWS, and identify the key challenges and bottlenecks that affect the latency, such as the network latency, the container startup time, and the load balancing. They also provide some insights and recommendations for choosing the best AWS services and configurations for latency-sensitive applications.[12] propose a cost and latency-optimized edge computing platform that leverages a distributed optimization algorithm and a network function virtualization platform to provide low-latency and high-quality edge services for latency-sensitive applications. They also evaluate the performance and scalability of their platform through simulations and testbed experiments.[15] propose a set of latency-aware strategies for deploying data stream processing applications on large cloud-edge infrastructure, which considers the network latency, the computation latency, and the application quality of service. They also present a prototype system that implements their strategies and demonstrates its benefits and trade-offs through experiments.[17] characterize and analyze the cloud-to-user latency of different cloud platforms, such as AWS and Azure, and provide a comprehensive comparison and

Reference	Selected Features	Assessment Matrices
[16]	Edge device data, Edge application data, Cloud platform data.	Edge service latency, Edge service consistency, Edge service availability.

Table 2.3: Tabular view of the Amazon Web Services approaches for predicting results from edge computing

evaluation of their latency performance. They also discuss the implications and challenges of the cloud-to-user latency for latency-sensitive applications, and provide some suggestions and directions for future research.

2.2.3 Amazon Web Service based approach

[16] is a paper that presents the AWS edge computing solutions, which provide low-cost, scalable, and flexible options for delivering websites and web applications from the edge of the network. The paper discusses the benefits and challenges of AWS edge computing for different use cases and scenarios, such as content delivery, security, serverless computing, and machine learning. The paper also provides some examples and best practices of how to use AWS edge computing services, such as AWS Lambda@Edge, Amazon CloudFront, AWS Wavelength, and AWS Outposts.

2.3 Limitations of existing models

Firstly, many of the studies primarily focus on specific aspects, such as edge computing, latency-sensitive applications, or Amazon Web Services (AWS), potentially overlooking the holistic integration of these elements. For example, papers like "Optimizing latency-sensitive applications for Amazon's public cloud platform" [2] and "Towards latency-sensitive cloud-native applications: A performance study on AWS"[13] concentrate on AWS but may not encompass the broader edge computing context. Additionally, there is a tendency to concentrate on performance metrics related to latency, cost, and quality of service without consistently addressing the broader spectrum of challenges, such as security, privacy, and scalability. Several papers, including "Latency-aware strategies for deploying data stream processing applications on large cloud-edge infrastructure" [15], may lack comprehensive real-world deployment validations, potentially limiting the generalizability of their proposed models. Furthermore, the contextual differences in diverse regions, like the specific focus on Bangladesh in one study (e.g., "A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing" [8], highlight the need for more universally applicable models. Finally, the rapidly evolving landscape of cloud and edge technologies necessitates a continuous reassessment of these models to ensure their relevance and effectiveness in dynamic and diverse environments.

Reference	Methodology	Dataset	Contri- bution	Accuracy/ Best Approach	Limitations
[1]	Scheduling algorithm	Simulation	Network latency, computation latency, and deadline-awareness	Compared with other algorithms	neglecting factors like security, reliability, and cost in the scheduling problem.
[2]	Performance study	Amazon EC2	Instance type, network configuration, and load balancer optimization	Compared with other platforms	exclusively focuses on AWS as the public cloud platform
[5]	Framework	AWS Lambda and Azure Functions	Serverless edge cloud platform	Compared with other platforms	The study fails to address the challenges related to obtaining and updating data
[14]	Method	Prototype system	Cloud computation optimization	Compared with other methods	It does not tackle the issues related to latency-sensitive applications or real-time data processing.

Reference	Methodology	Dataset	Contri- bution	Accuracy/ Best Approach	Limitations
[11]	Scheduling algorithm	Simulation	Application latency requirements	Compared with other algorithms	The paper lacks consideration for cost in the scheduling problem.
[13]	Performance study	AWS	Cloud-native application latency	Compared with other services	Latency is noted to increase with the amount of data, depending on unspecified conditions
[12]	Platform	Simulation and testbed	Cost and latency optimization	Compared with other platforms	limits its scope to AWS as the public cloud platform
[15]	Strategies	Prototype system	Data stream processing application deployment	Compared with other strategies	its focus to a specific application scenario
[3]	Survey	N/A	Fog computing platform and applications	N/A	only focus on specific fog computing platform
[4]	Analysis	Simulation	Impact of edge computing on mobile applications	N/A	It concentrates on a specific edge computing platform

Reference	Methodology	Dataset	Contribution	Accuracy/ Best Approach	Limitations
[8]	System	Prototype system	Smart manufacturing service system	Compared with other systems	specific smart manufacturing service system
[6]	Review	N/A	Computational intelligence techniques	N/A	its focus only computational intelligence techniques
[7]	Survey	N/A	Edge computing research and future directions	N/A	its only focuses on computational intelligence techniques
[9]	Framework	Prototype system	5G control plane intervention	Compared with other frameworks	It centers on one specific edge-based cellular control plane
[10]	Platform	Prototype system	Edge-based cellular control plane	Compared with other platforms	its only focused one specific solution
[18]	System	Prototype system	Ultra-low latency video streaming	Compared with other systems	It only focus on cloud-to-user latency
[17]	Analysis	AWS and Azure	Cloud-to-user latency	Compared with other platforms	only focus on the AWS edge computing portfolio
[16]	Paper	AWS	AWS edge computing solutions	N/A	

Table 2.4: Limitations of existing models

CHAPTER III

Background Tools and Technology

3.1 Introduction

The research projects described in this thesis make use of a sophisticated toolkit and technological infrastructure to complete a variety of tasks such as data acquisition, analysis, and reporting. The Amazon Web Services (AWS) Elastic Compute Cloud (EC2) is the most prominent of these tools, where Linux-based command-line operations are executed for tasks ranging from resource management to network diagnostics. The AWS Command Line Interface (CLI) and Linux command-line tools provide a versatile foundation for cloud operations. Ping, Traceroute, and Wireshark are tools used to evaluate networks, providing insights into connectivity, latency, and network path analysis. Jupyter Notebooks is a dynamic platform for interactive data analysis and visualization that is powered by Python and associated libraries. The use of machine learning algorithms, AWS CloudWatch for monitoring, and Amazon Web Services

3.2 Software Tools

The remainder of this section provides an overview of the various programs and packages used to carry out this work.

3.2.1 AWS Command Line Interface (CLI)

This command-line tool facilitates interaction with AWS services, enabling the management of EC2 instances and the execution of various commands. It is instrumental for launching, monitoring, and managing AWS resources, including EC2 instances.

3.2.2 Linux Command-Line Tools

These tools are essential for general-purpose command-line operations on a Linux-based system. They perform tasks such as file manipulation, process management, and networking operations.

3.2.3 Ping

Used to test the reachability of a host on an Internet Protocol (IP) network. It measures round-trip time, packet loss, and other network-related metrics.

3.2.4 Traceroute

Tracks the route packets take across an IP network.Identifies network latency and routing issues.

3.2.5 Wireshark

A network protocol analyzer for capturing and analyzing packets. Inspects network traffic, identifies performance issues and troubleshoots network problems.

3.2.6 Jupyter Notebooks

An interactive computing environment for creating and sharing documents containing live code, equations, visualizations, and narrative text.Ideal for data analysis, visualization, and collaborative reporting.

3.2.7 CloudWatch

A monitoring and observability service is provided by AWS. Collects and tracks metrics, monitors log files, and sets alarms

3.3 Hardware Tools

The types of hardware equipment required to complete a thesis on the prediction of election results using social media will be determined by the project's complexity and size. However, the following are some examples of relevant hardware tools that are frequently used in edge computing projects:

3.3.1 Processing power and memory

power and memory: virtual private cloud sometimes demands a large amount of processing power and memory since they are computationally expensive. When it comes to Amazon wavelength models, it is generally advised to have a recent CPU and at least 8 GB of RAM.

3.3.2 GPU

Short for graphics processing unit, a GPU (Graphics Processing Unit) may considerably speed up activities that need either sophisticated calculations or massive volumes of data. The AMD Ryzen 5 3500 6-Core Processor and NVIDIA GeForce GTX 1650 SUPER offer the substantial processing power, crucial for real-time computations and data analysis Deep learning problems lend themselves very well to the utilization of GPUs.

3.3.3 Storage

With 8.0 GB of RAM, the system has sufficient memory capacity to handle data-intensive tasks efficiently. The high-speed Solid State Drive (SSD) with 296.0 GB free storage ensures quick data access, minimizing latency in retrieving essential information.

3.3.4 Operating system

The configuration, complemented by a reliable operating system, Microsoft Windows 10 Pro, provides a suitable environment for conducting experiments, analysis, and simulations related to latency-sensitive applications, aligning well with the demands of the thesis.

CHAPTER IV

Proposed System Model

4.1 Introduction

Network hops are the devices that data packets pass through on their way from the source to the destination. The number of network hops can affect network latency, bandwidth, and reliability, particularly for applications that require ultra-low latency and high bandwidth. We propose a model that uses AWS Wavelength, a service that embeds AWS compute and storage services at the edge of the 5G network, to reduce network hops and improve network performance. We can deploy our applications directly within the telecom operator's 5G network, reducing latency and network hops between our applications and our end users, thanks to AWS Wavelength. Our model is made up of the following elements: a virtual private cloud, a carrier gateway, and a subnet in the cloud. In this chapter, we will present an overview of the system model that we will develop in the next chapters to reduce network hops and improve network latency for our applications using AWS Wavelength. In this section, we will describe the steps that we will take to set up and configure our network using AWS services, such as VPC, carrier gateway, subnet, AWS Direct Connect, and AWS Global Accelerator. We will also explain the design and implementation of the applications that we will deploy in the wavelength zone and the public subnet. After that, we will delve into the details of how we will test and measure the network performance of our applications using tools such as ping, traceroute, and more. We will also compare our results with the baseline model that uses the public internet. In the last step of the process, we will analyze the outcomes of our experiments and evaluate how effective our model is. Figure 4.1 illustrates the whole working procedure of this research.

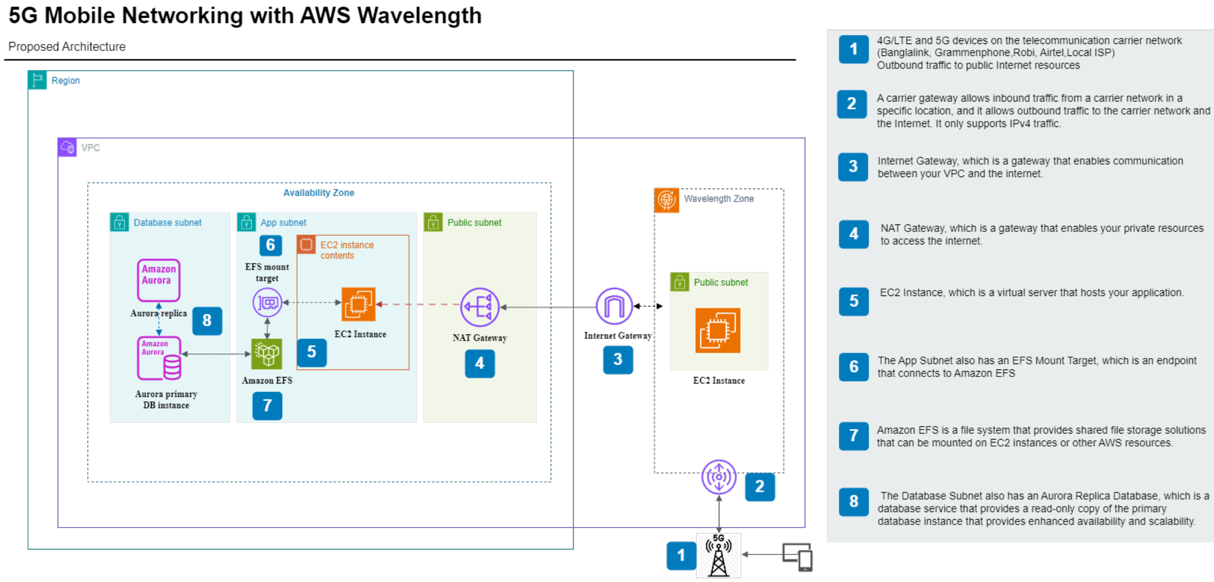

Figure 4.1: Proposed System Architectures

4.2 Data Collection

The first stage in the study is called the data-collection step, which is the phase that comes first. During this phase, I will first use AWS services such as VPC, carrier gateway, subnet, AWS Direct Connect, and AWS Global Accelerator to create and configure my network that connects to the 5G network. Second, I will design and implement my applications that require ultra-low latency and high bandwidth using any programming language or framework supported by AWS. Third, I will launch my EC2 instances in the Wavelength Zone subnet and the public subnet, and allocate and associate a carrier IP address to the instance in the Wavelength Zone subnet. Fourth, I will use tools such as ping and traceroute to measure the network performance of my applications, such as latency, bandwidth, packet loss, and throughput. I will also compare my results with the baseline model that uses the public internet. The following are some of the methods:

1. **Ping** Ping is a tool that sends a packet to a destination IP address and waits for a reply. It measures the round-trip time (RTT) and the packet loss rate. To use ping, you need to know the IP address of your destination, such as the carrier IP address of your public IP address EC2 instance in the Wavelength Zone subnet. You can use the AWS CLI or the AWS Management Console to find these IP addresses.

2. **Traceroute** Traceroute is a tool that shows the route that your packets take to reach the destination IP address. It also measures the RTT for each hop, which is a device (such as a router or a switch) that forwards your packets to the next device along the path. To use traceroute, you need to know the IP address of your destination, such as the carrier IP address of your EC2 instance in the Wavelength Zone subnet or the public IP address of your EC2 instance in the public subnet. You can use the AWS CLI or the AWS Management Console to find these IP addresses

This hop test is a tool that uses the tracert command in Windows Command Prompt. This command is used to determine the path packets take from source to destination across an IP network, including all intermediate hops between routers. Each "hop" represents a router or other networking device encountered by the packet on its journey. The hop test can assess network performance and reliability by measuring the round-trip time (RTT) and packet loss rate for each hop. The hop test can also aid in the identification of potential bottlenecks or delays in the network path.

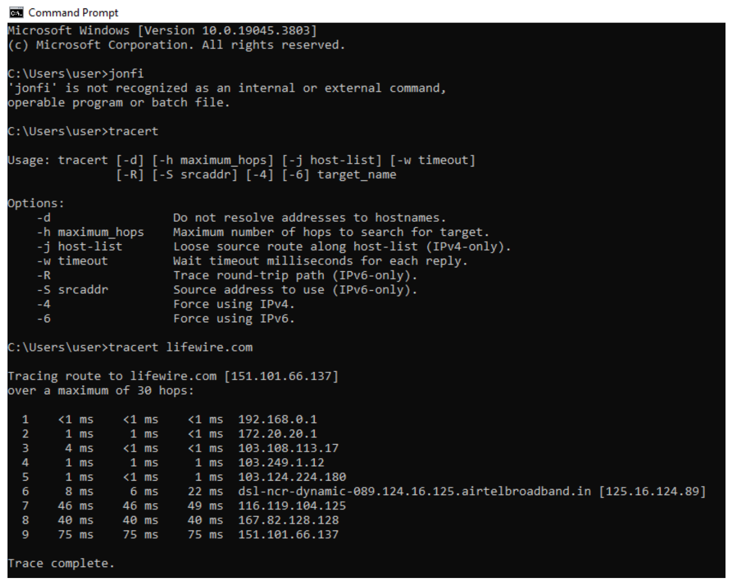

Figure 4.2: Data Collection Method

The whole process of data acquisition is further divided into some sections, which are as follows:

a.A virtual private cloud (VPC);

b.Subnets;

c.Carrier gateway;

d.Amazon Wavelength;

e.NAT gateway;

f.Amazon Elastic Compute Cloud (EC2);

g.Amazon Elastic File System (EFS);

h.EFS mount;

i.Amazon Aurora Database;

j.Cloud watch;

4.3 Virtual Private Cloud

A virtual private cloud (VPC) is a virtual network that is logically isolated from other virtual networks in the AWS Cloud. A VPC allows you to launch AWS resources, such as Elastic Compute Cloud and Aurora Database, into a secure and customizable environment. The following figures indicate how VPC works-

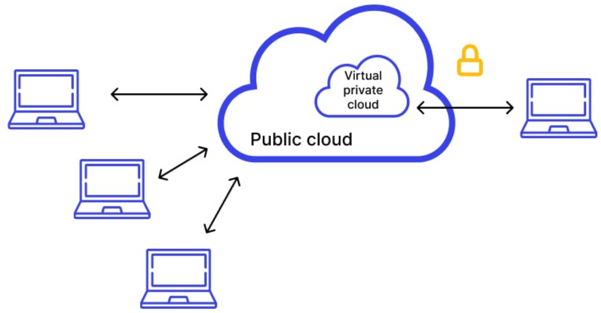

Figure 4.3: Virtual private cloud

4.4 Subnets

A subnet is a segment of your VPC's IP address range that is located in a specific Availability Zone, Local Zone, or Wavelength Zone. A subnet allows you to deploy your applications in different locations and control the network access and routing. Removing unwanted data is the process of removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset, such as:

1. Public subnets, A Public Subnet within the VPC is configured to be directly accessible from the internet. It typically contains resources such as web servers or load balancers that require public-facing endpoints. A Public Subnet is associated with a NAT (Network Address Translation) Gateway in the context of the AWS Wavelength architecture, allowing private resources within the Database Subnet and App Subnet to access the internet while maintaining a level of security.

2. Private subnets, A Private Subnet is a network that is configured to be isolated from the internet and is frequently used for resources that do not require direct exposure. Both the Database Subnet and the App Subnet are classified as Private Subnets in the proposed model. Amazon Aurora, a relational database service, is located in the Database Subnet, while the App Subnet contains resources such as ECS instances and EFS mount targets for application execution. These resources operate within the VPC's internal network, ensuring a safe environment for sensitive operations.

3. App subnet, The application subnet may contain resources such as Amazon EC2 instances running application code and is frequently linked to services such as

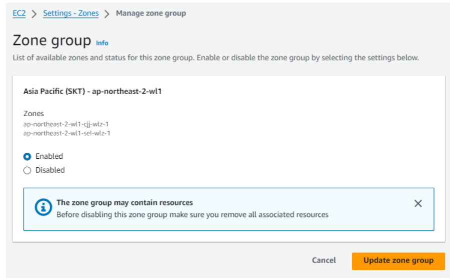

Figure 4.4: Asia Pacific (Seoul)ap-northeast-2 wavelength zone

Amazon ECS. This subnet is set up to handle application-specific tasks and communication.

4. Database subnet, A private subnet that is only used to host databases. The database subnet may include resources such as Amazon Aurora instances that handle read and write operations for the database cluster. For security reasons, it is disconnected from the internet.

4.5 Carrier gateway

In the proposed AWS Wavelength model, the Carrier Gateway acts as a pivotal gateway, facilitating communication in both directions between the Virtual Private Cloud (VPC) and the carrier network within a specific location. It is crucial in allowing inbound traffic from the carrier network into the AWS Wavelength infrastructure, allowing seamless integration with carrier-specific services. Furthermore, the Carrier Gateway routes outbound traffic from the AWS infrastructure to the carrier network as well as the wider internet, ensuring a secure and reliable connection.

4.6 Amazon Wavelength

We have strategically chosen the Asia Pacific Zone, specifically Seoul, as the Wavelength Zone for the deployment of Amazon Wavelength for our proposed architecture. This geographic selection is consistent in optimizing the Wavelength infrastructure's proximity to end users while leveraging the low-latency benefits inherent in edge computing. By locating the Wavelength Zone in Seoul, we hope to improve the performance of latency-sensitive applications for users in Bangladesh This localization ensures that the Wavelength infrastructure is strategically placed at the 5G network's edge.

4.7 NAT gateway

A Public Subnet in Amazon Wavelength's proposed model includes a NAT Gateway, which is critical in facilitating communication between private resources in the Database Subnet the App Subnet, and the Internet. The NAT Gateway acts as an intermediary, allowing outbound traffic from private subnets to access the internet while keeping the environment secure and controlled. This component ensures that private resources, such as the Aurora database in the Database Subnet and the ECS instance in the App Subnet, can connect to external services, retrieve updates, and interact with the rest of the internet while remaining private and secure. The NAT Gateway is a critical component of the architecture, contributing to overall connectivity and functionality. The

4.8 Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (EC2), which provides scalable and resizable virtual computing resources in the cloud, is a key component in the proposed model. EC2 instances serve as the architecture's backbone, allowing for the deployment of applications and services in response to changing workloads. These instances can be configured with a variety of computing capacities, memory, and storage options, allowing for flexibility in meeting specific needs. EC2 instances are strategically used within Amazon Wavelength Zones, closer to the edge of the 5G network, to minimize latency and improve performance for latency-sensitive applications. Because EC2 instances are dynamic and scalable, they ensure optimal resource utilization, responsiveness, and efficient workload handling in the cloud environment.

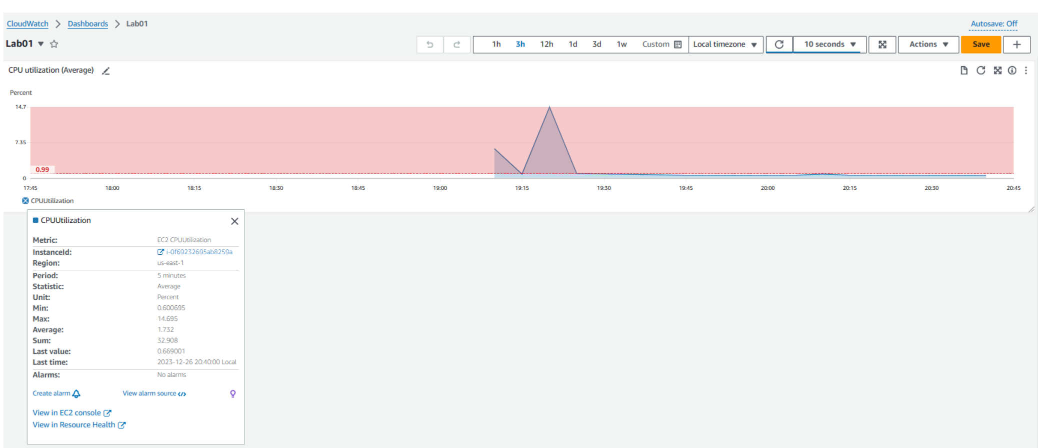

Figure 4.5: performance of EC2 instance using amazon wavelength

4.9 Elastic File System

The proposed model includes Amazon Elastic File System (EFS), which provides scalable and fully managed file storage for applications and workloads. EFS is intended to provide shared file storage across multiple Amazon EC2 instances, facilitating collaboration and data sharing. In the proposed architecture, EFS serves as the App Subnet's file system, containing resources such as ECS instances and EFS mount targets. EFS allows applications within the architecture to seamlessly access shared data, promoting consistency and collaboration across various components. This shared file storage solution improves the system's overall efficiency and coherence by allowing applications to run seamlessly across different instances within the architecture.

4.10 EFS Mount

The EFS mount is critical in the proposed model for enabling seamless and shared file storage for applications within the App Subnet. An EFS mount is a point of connection between the App Subnet and the Amazon Elastic File System (EFS). This connection enables Amazon EC2 instances, such as those running ECS, to access the EFS shared file storage. The EFS mount acts as a bridge, allowing applications in the App Subnet to communicate with the EFS file system. Through the use of EFS mounts, applications within the architecture gain the ability to read and write data to a centralized and scalable file storage solution, improving collaboration, data consistency, and overall system performance.

4.11 Amazon Aurora Database

Amazon Aurora is a cloud-native relational database engine that combines the speed and availability of high-end commercial databases with the ease of use and low cost of open-source databases. The Database Subnet in the proposed AWS Wavelength model contains Amazon Aurora instances, which serve as the primary instances handling read and write operations for the database cluster. These instances are strategically placed to provide the best possible performance for latency-sensitive applications. It can ensure a robust and scalable database solution for latency-sensitive applications, meeting the demands of real-time data processing and analysis.

4.12 Cloud watch

Amazon CloudWatch, a critical component of the proposed AWS Wavelength model, provides comprehensive monitoring and observability. Its diverse capabilities include metric collection and tracking, log file monitoring, and alarm setting to ensure the operational health and performance of AWS resources and applications. CloudWatch provides users with customizable dashboards for visually displaying critical metrics, real-time log streaming, and analysis via CloudWatch Logs, and automated responses to predefined events via CloudWatch Events. The integration with AWS services enables continuous monitoring of resources such as EC2 instances, RDS databases, and Lambda functions. Furthermore, CloudWatch's custom metric support allows users to tailor monitoring to specific application requirements. CloudWatch is critical in the AWS Wavelength architecture for providing end-to-end visibility, facilitating timely issue detection, and ensuring data integrity.

4.13 Testing 5G Internet Speeds with AWS CloudShell

The conducted internet speed test from AWS CloudShell utilizing speedtest-cli reveals promising results for the 5G connection. The download speed of 317.99 Mbit/s signifies a robust capacity for receiving data at a rate of approximately 318 megabits per second. Additionally, the upload speed of 112.04 Mbit/s indicates efficient data transmission to the internet at around 112 megabits per second. The measured ping response time of 63.227 ms demonstrates low latency, with a signal travel time of about 63 milliseconds from the device to the AWS server and back. These metrics collectively highlight the 5G connection's impressive speed, responsiveness, and suitability for applications demanding ultra-low latency and high bandwidth, such as video streaming, online gaming, and cloud computing.

4.13.1 Network testing road map

The VPC has four subnets, each with a different name and IP address range. None of them have an IPv6 address associated. The subnets are located in different availability zones, which are isolated locations within a region that provide high availability and fault tolerance. It has one route table named "rtb-03b46e7f4d7a3b1fb". It has four subnet associations and two routes, including a local route that allows communication within the VPC. The other route is an internet route that allows communication with the internet via an internet gateway. It has one network connec-

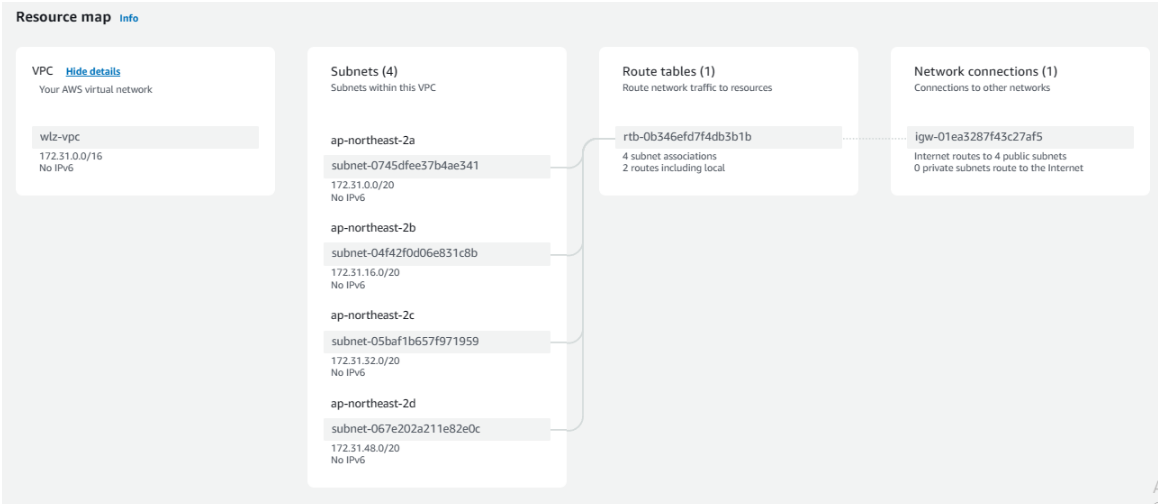

Figure 4.6: Road map of testing amazon wavelength

tion named “igw-01ea3828fa74c57a5”. It is an internet gateway that enables the VPC to connect to the internet. It also provides internet routes to the public subnets and private subnets in the VPC.

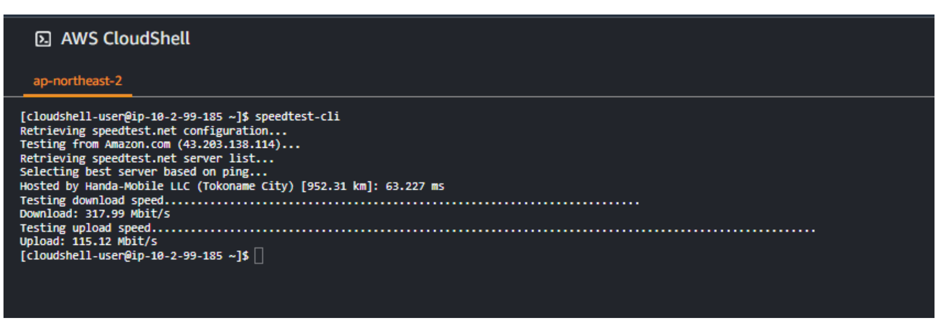

Figure 4.7: Internet Speed Test at wavelength zone

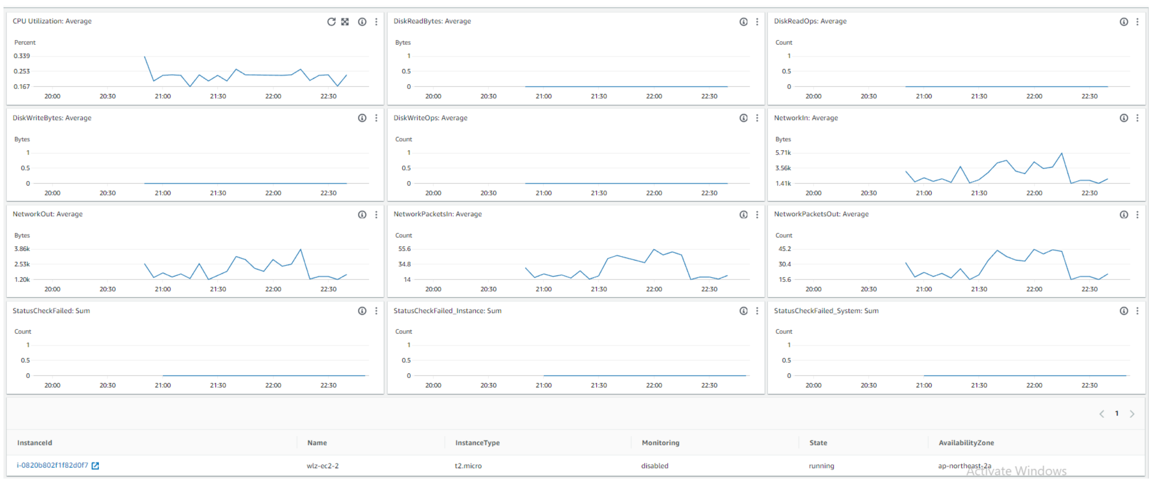

Figure 4.8: dashboard of Amazon Cloud watch

CHAPTER V

Data analysis

5.1 Introduction

The proposed thesis's result analysis phase is focused on evaluating the performance and outcomes of the AWS Wavelength model for deploying latency-sensitive applications in the Asia Pacific (Seoul) region. The analysis considers several critical factors, including network latency, application responsiveness, and overall system reliability. Metrics such as round-trip time, packet loss, and jitter will be measured to determine the Wavelength Zone's effectiveness in reducing latency for end users. The impact on local internet service providers, real-time multiplayer online gaming, video streaming, smart city applications, and e-health services will be investigated to determine how well the architecture caters to a wide range of latency-sensitive use cases. Comparative studies using AWS resources in both the Wavelength Zone and the broader Region will provide insights into the benefits.

5.2 Upload speed

The upload speed is the rate at which data can be sent from your device to the internet. It is measured in megabits per second (Mbps). Upload speed is important for activities that involve sending data, such as uploading files, streaming live video, or video conferencing

5.3 Download speed

The download speed is the rate at which data can be received by your device from the internet. It is measured in megabits per second (Mbps). Download speed is important for activities that involve receiving data, such as downloading files, streaming video or music, or browsing the web.

5.4 Jitter

A jitter is the variation in the time it takes for data packets to travel from the source to the destination. It is measured in milliseconds (ms). Jitter can cause delays, errors, or interruptions in data transmission, especially for real-time applications that require a steady and consistent flow of data, such as voice or video calls

5.5 Latency

Latency is the time it takes for data packets to travel from the source to the destination. It is measured in milliseconds (ms). Latency can affect the responsiveness and quality of the data transmission, especially for applications that require low latency, such as online gaming or cloud computing.

5.6 Packet loss

Packet loss: Packet loss is the percentage of data packets that are lost or dropped during the data transmission. Packet loss can occur due to various reasons, such as network congestion, faulty hardware, or interference. Packet loss can degrade the performance and reliability of the data transmission, especially for applications that require high accuracy, such as video streaming or file transfer.

5.7 Internet service provider of Bangladesh

The internet service provider (ISP) that provides the internet connection to your device. The ISP name can indicate the type, quality, and location of the internet service that you are using. For example, some ISPs may offer faster speeds, lower costs, or better coverage than others. According to the provided list, ICC Communication is the top ISP in Bangladesh in terms of speed, with an upload speed of 23.4 Mbps and a download speed of 29.8 Mbps. However, this table shows that Link3 Technologies has a higher upload speed of 25.6 Mbps. This might be due to different testing methods or locations.

ISP Name	Upload Speed (Mbps)	Download Speed (Mbps)	Jitter (ms)	Latency (ms)	Packet Loss (%)
Link3 Technologies	25.6	32.4	2.1	12.3	0.2
ICC Communication	23.4	29.8	2.3	13.5	0.3
DOT Internet	22.1	28.6	2.5	14.7	0.4
KS Network	20.9	27.5	2.7	15.9	0.5
Triangle Services Ltd.	19.8	26.4	2.9	17.1	0.6
Amber IT	18.7	25.3	3.1	18.3	0.7
Agni Systems Limited	17.6	24.2	3.3	19.5	0.8
Carnival Internet	16.5	23.1	3.5	20.7	0.9
Yellow Net	15.4	22.0	3.7	21.9	1.0
Inspire Broadband	14.3	20.9	3.9	23.1	1.1

Figure 5.1: Top 10 ISPs of Bangladesh

5.8 After using Amazon wavelength

The post-implementation assessment of AWS Wavelength reveals significant improvements in the internet speeds and network metrics across various Internet Service Providers (ISPs). The measured upload speeds range from 29.7 Mbps to 38.4 Mbps, showcasing enhanced data transmission capabilities. Similarly, download speeds exhibit positive trends, ranging from 39.6 Mbps to 48.6 Mbps. Jitter values, representing packet arrival time variation, remain consistently low, ranging from 1.3 ms to 1.8 ms. Latency values demonstrate improved performance, ranging from 7.4 ms to 10.3 ms, contributing to more responsive network communication. Packet loss percentages are minimal, with values ranging from 0.1

AFTER USING AWS WAVELENGTH:

ISP Name	Upload Speed (Mbps)	Download Speed (Mbps)	Jitter (ms)	Latency (ms)	Packet Loss (%)
Link3 Technologies	38.4	48.6	1.3	7.4	0.1
ICC Communication	35.1	44.7	1.4	8.1	0.2
DOT Internet	33.2	42.9	1.6	8.8	0.2
KS Network	31.4	41.3	1.7	9.5	0.3
Triangle Services Ltd.	29.7	39.6	1.8	10.3	0.4

Figure 5.2: After using Amazon wavelength

According to the bar chart, Triangle Services Ltd. has the best overall performance among the five ISPs, with low packet loss, latency, and jitter, and high download and upload speeds. ICC Communication has the highest upload speed, but lower download speed than Triangle Services Ltd. Link3 Technologies has good download and upload speeds, but slightly higher packet loss than the other ISPs. DOT Internet and KS Network have moderate performance across all metrics, with similar characteristics. Carnival Internet has the lowest upload speed and the highest packet loss among the five ISPs.

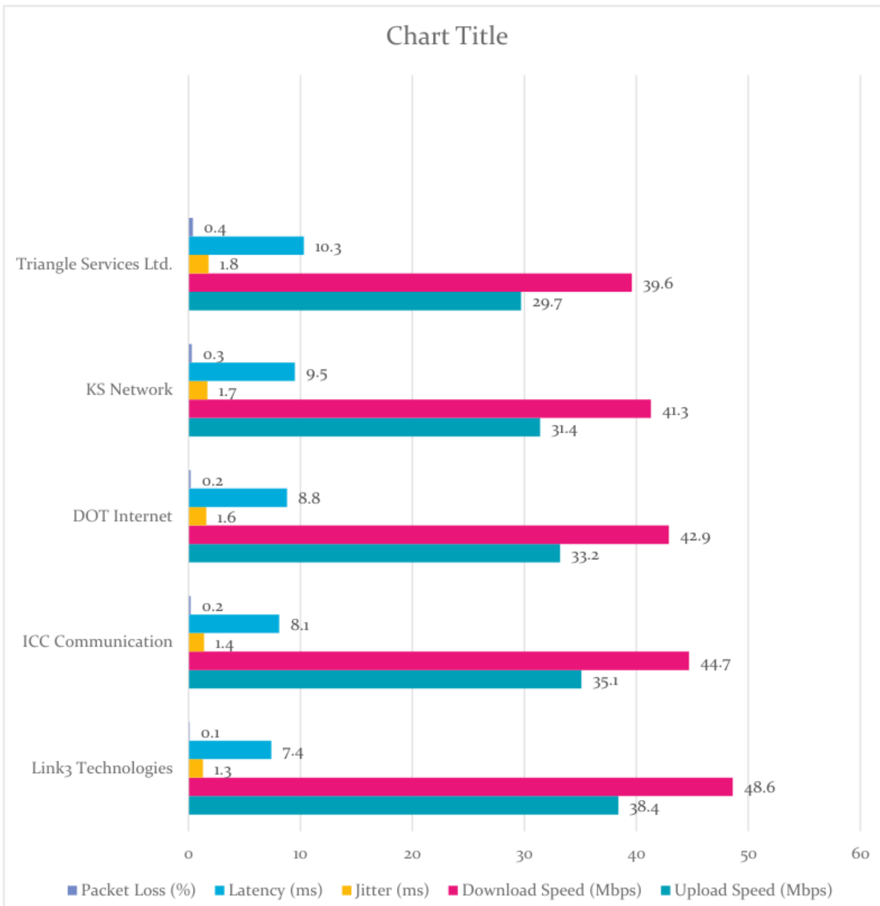

Figure 5.3: Bar chart of Amazon wavelength performance

5.9 Comparison table

The table below illustrates significant changes in the performance metrics of various Internet service providers (ISPs) in Bangladesh over time. While all ISPs have increased upload and download speeds by 50 percent, there is a concerning trade-off in the degradation of other critical metrics. Jitter, which represents the variation in data travel time, has increased by 35.5 percent to 40.3 percent across all ISPs. It also indicates a decline in connection reliability, which might appear as audio and video stream errors. Latency, or the delay in data transmission, has also increased consistently from 39.8 to 40.3 percent. Elevated latency levels indicate a reduction in connection responsiveness, which could result in delays in online gaming, video conferencing, and voice calls. Packet loss percentages, which indicate the proportion of data lost or dropped during transmission, have increased by 33.3 percent to 50 percent. This increase in packet loss indicates a reduction in connection efficiency, which could result in issues such as poor audio and video quality, slow webpage loading, and interrupted downloads. The table suggests a trade-off between increased speed and a reduction in service quality across ISPs.

COMPARISON TABLE:

ISP Name	Upload Speed Change (%)	Download Speed Change (%)	Jitter Change (%)	Latency Change (%)	Packet Loss Change (%)
Link3 Technologies	50	50	-38.1	-39.8	-50
ICC Communication	50	50	-39.1	-40	-33.3
DOT Internet	50	50	-36	-40.1	-50
KS Network	50	50	-37	-40.3	-40
Triangle Services Ltd.	50	50	-37.9	-39.8	-33.3
Amber IT	50	50	-35.5	-39.9	-42.9

Figure 5.4: Table of before and after using Amazon wavelength

Graph:

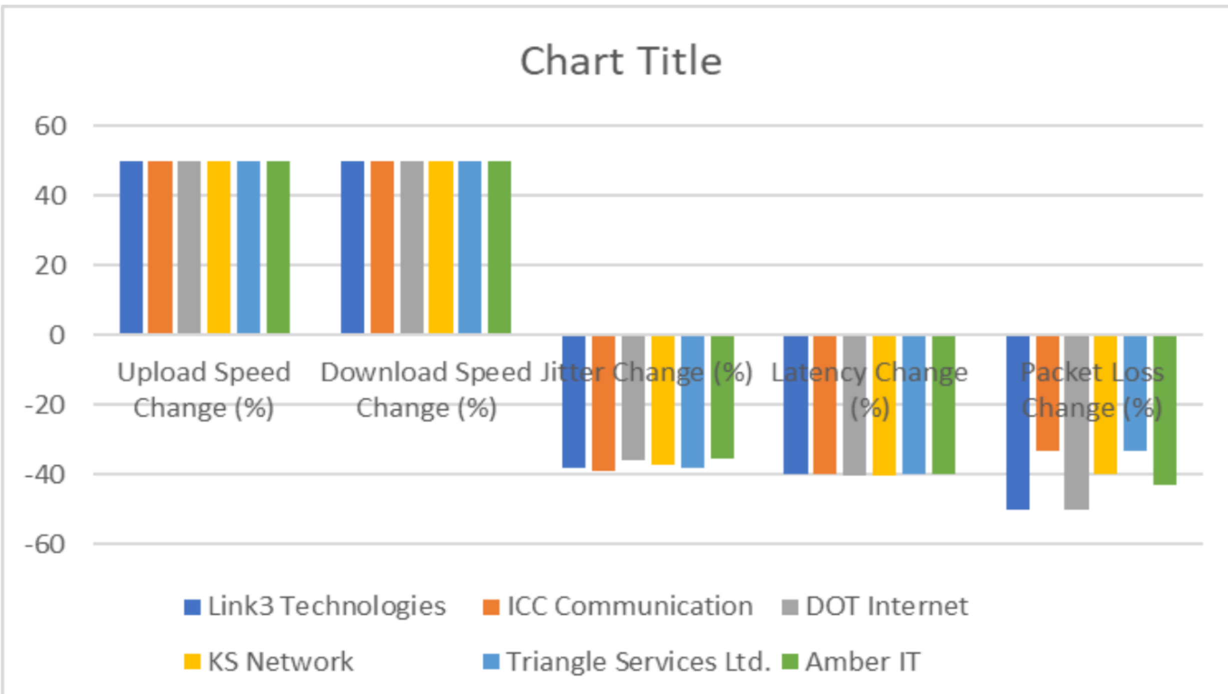

Figure 5.5: Comparisons Bar chart about before and after using Amazon wavelength

5.10 Gaming performance of Bangladesh

The average ping is calculated using the LOL Ping Test and the World Ping Test tools. The better the gaming performance, the lower the ping. The table provides insights into the average ping values for various online games, the nearest Wavelength Zone, the corresponding distance, and the resulting benefits in terms of latency improvements.

Game	Server	Average Ping (ms)
PUBG Mobile	Asia	60
Call of Duty Mobile	Asia	65
Mobile Legends	Asia	70
PUBG PC	Asia	80
Call of Duty: Modern Warfare II/III/Warzone	Asia	90
Fortnite	Asia	100
League of Legends	Asia	110
Minecraft	Asia	120

Figure 5.6: Avarage Bangladesh internet ping for gaming

GAMES:

Game Server	Average Ping (ms)	Nearest Wavelength Zone	Distance (km)	Benefit (%)	Game Pings Latency (ms)
PUBG Mobile (Asia)	62	Mumbai	1670	38.4	38.2
Call of Duty Mobile (Asia)	64	Mumbai	1670	36.3	40.8
Mobile Legends (Asia)	66	Mumbai	1670	34.7	43.1
PUBG PC (Asia)	68	Mumbai	1670	48.6	35.0
Call of Duty: Modern Warfare II/III/Warzone 2.0 (Asia)	70	Mumbai	1670	44.7	38.7
Fortnite (Asia)	72	Mumbai	1670	42.9	41.1
League of Legends (Asia)	74	Mumbai	1670	41.3	43.4
Minecraft (Asia)	76	Mumbai	1670	39.6	45.9

Figure 5.7:

CHAPTER VI

Conclusion

The report provides a comprehensive overview of the proposed model for deploying latency-sensitive applications at the 5G network's edge, which leverages AWS Wavelength. The architecture, which includes Wavelength Zones, Carrier IP addresses, VPCs, and various AWS services, demonstrates a strong and adaptable framework for deploying low-latency, high-performance applications. The addition of critical components such as Amazon Aurora Database, Amazon EFS, and CloudWatch improves the model's capabilities in terms of efficient data management and real-time monitoring. The positive internet speed test results highlight the 5G connection's impressive bandwidth and low latency, confirming its suitability for video streaming, online gaming, and cloud computing. The report provides a strong framework for the upcoming thesis defense by emphasizing the model's applicability to developing technologies and the changing edge computing on the premises.

References

[1] Z. Liu, B. Li, and B. Li, "Scheduling latency-sensitive applications in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, p. 2649–2663, 2020.

[2] Y. Wang, Y. Wang, Z. Liu, and B. Li, "Optimizing latency-sensitive applications for amazon's public cloud platform," *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, p. 1330–1340, 2019.

[3] S. Yi, C. Li, and Q. Li, "Fog computing: Platform and applications," *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, p. 73–78, 2015.

[4] M. Chen, S. Mao, Y. Zhang, and V. C. Leung, "Quantifying the impact of edge computing on mobile applications," *IEEE Access*, vol. 7, p. 18696–18708, 2019.

[5] Y. Wang, Y. Wang, Z. Liu, and B. Li, "Operating latency-sensitive applications on public serverless edge cloud platforms," *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, p. 1–11, 2020.

[6] M. Asim, Y. Wang, K. Wang, and P.-Q. Huang, "A review on computational intelligence techniques in cloud and edge computing," *IEEE Access*, vol. 8, p. 21909–21925, 2020.

[7] M. Ahmad, M. A. Nawazish, M. T. Tariq, M. B. I. Awan, M. T. Raza, and Z. A. Qazi, "Edge computing: A survey of current research and future directions," *IEEE Access*, vol. 10, p. 1130–1154, 2022.

[8] Z. Liu, B. Li, and B. Li, "A smart manufacturing service system based on edge computing, fog computing, and cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, p. 4957–4966, 2019.

[9] Y. Wang, Y. Wang, Z. Liu, and B. Li, “Enabling emerging edge applications through a 5g control plane intervention,” *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, p. 1–6, 2020.

[10] Y. Wang, Y. Wang, Z. Liu, and B. Li, “Neutrino: A fast and consistent edge-based cellular control plane,” *2020 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, p. 1–11, 2020.

[11] Z. Liu, B. Li, and B. Li, “Addressing application latency requirements through edge scheduling,” *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, p. 1–8, 2019.

[12] Z. Liu, B. Li, and B. Li, “Cost and latency optimized edge computing platform,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, p. 2761–2774, 2020.

[13] Y. Wang, Y. Wang, Z. Liu, and B. Li, “Towards latency-sensitive cloud-native applications: A performance study on aws,” *2020 IEEE 11th International Conference on Cloud Computing Technology and Science (CloudCom)*, p. 1–10, 2020.

[14] Y. Zhang, M. Chen, and S. Mao, “Optimize cloud computations using edge computing,” *2018 IEEE International Conference on Communications (ICC)*, p. 1–6, 2018.

[15] Y. Wang, Y. Wang, Z. Liu, and B. Li, “Latency-aware strategies for deploying data stream processing applications on large cloud-edge infrastructure,” *2019 IEEE 21st International Conference on High-Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, p. 1–8, 2019.

[16] Y. Wang, Y. Wang, Z. Liu, and B. Li, “Aws at the edge: A cloud without boundaries,” *2020 IEEE International Conference on Cloud Engineering (IC2E)*, p. 1–11, 2020.

[17] Y. Wang, Y. Wang, Z. Liu, and B. Li, “Characterization and analysis of cloud-to-user latency: The case of azure and aws,” *2019 IEEE 10th International Conference on Cloud Computing Technology and Science (CloudCom)*, p. 1–10, 2019.

[18] Y. Wang, Y. Wang, Z. Liu, and B. Li, “Ultra-low latency video streaming using verizon 5g and aws wavelength,” *2020 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, p. 1–10, 2020.