REAL-TIME STREAMING PROTOCOL USING LINEAR REGRESSION

A thesis submitted in partial fulfilment of the requirements for the award of the degree of

B.Tech in
Computer Science and Engineering

By

Khushi Goel (106121064)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY TIRUCHIRAPPALLI - 620015

NOVEMBER 2023

REAL-TIME STREAMING PROTOCOL USING LINEAR REGRESSION

ABSTRACT

a) Importance in Research:

In the research area, Real Time Streaming Protocol plays a significant role in streaming multimedia content, it basically focuses on facilitating successful communication between client and server.

b)Issues in this topic:

High Latency and more CPU utilization during Playback cause breaking of immersion of the user, and reduce the overall quality of the multimedia while playing.

c)Impact of Issues/Disadvantages:

A user may experience hindrances because of high latency which directly affects the quality of multimedia content and reduces the real-time immersion of the user.High Latency directly affects the CPU utilization which is also considered one of the drawbacks as it may cause slowing down the server.

d)Our Research to remove the current disadvantage

According to the research and by surfing various websites, we came to the conclusion of applying a Machine Learning algorithm -Linear Regression which will help us lower the latency and CPU utilization

e)Research Gaps:

Even though the research has already addressed various aspects of the RTSP, there is still a bridge that needs to be built for effortless streaming of multimedia content, The High latency and CPU utilization being the major.

f)Proposed Algorithm:

Our proposed algorithm aims to address the gap in existing works. Linear Regression works on optimizing the number of packets sent and received to minimize the loss and to achieve low latency and minimalistic use of CPU resources

g)Parameter improvement (in the proposed algorithm):

Our algorithms focus on improving parameters like Latency and CPU utilization reliability and reducing the loss of the number of packets in order to improve the performance of multimedia that is being streamed.

h)Research results:

According to the modification made by our research, we got the results as the playing video having low latency, these results are better than the existing works (base paper)

LINEAR REGRESSION

a) Our proposed algorithm

Our algorithm aims to optimize RTSP while keeping the major focus on keeping the latency of the video playing low and at least possible CPU utilization, algorithm introduces two approaches: LOW LATENCY AND LOW CPU UTILIZATION.

b) NOVELTY/CONTRIBUTION:

1. Smart Latency Reduction:

It uses machine learning techniques to reduce the delay in video, which ensures the user experiences comparatively less lag.

2.Efficient CPU Usage:

The aim is to use the resources of the CPU effectively for effortless streaming instead of increasing the wait time which may lead to deadlocks and starvation. Ineffective use of CPU resources may lead to slowing down your system.

3. Optimization:

Our proposed algorithm ensures the machine learning technique is combined with existing works in order to provide an output that results in low-lag video and not hamper your system.

c)INTRODUCTION TO THE WORK:

RTSP plays a very crucial role in multimedia streaming considering the increased rates of usage by users, the high latency and strain on our systems remains the same challenge to be addressed. Our proposed algorithm works on these issues to provide a better user experience.

d)MODULES

1.pandas

2.numpy

3.matplotlib

4.scikitlearn

e)INTRODUCTION TO MODULES :

1. pandas (import pandas as pd): used for data manipulation and analysis.

2. numpy (import NumPy as np): NumPy is used for numerical operations in Python.

3. matplotlib (import matplotlib. pyplot as plt) In this code, it is used for creating a scatter plot and regression line.

f) STEP BY STEP INSTRUCTIONS:

1. Importing Libraries

Step1: The necessary libraries/modules are imported.

2. Linear Regression Function

This function takes the path to a CSV dataset file as input and performs the following steps:

Load Dataset: Read the dataset from the CSV file using pd.read_csv.

Feature and Label Extraction: Assuming the dataset has specific columns, extract features ('SequenceNumber', 'Latency', 'CPUUtilization') and labels ('Label').

Train-Test Split: Split the dataset into training and testing sets using train_test_split.

Model Training: Train the model using the training data (X_train and y_train).

Make Predictions: Use the trained model to make predictions on the test set (X_test).

Model Evaluation: Evaluate the model's performance using accuracy and confusion matrix.

3. Main Execution

In the main execution block:

The path to the dataset file ('dataset.csv') is provided.

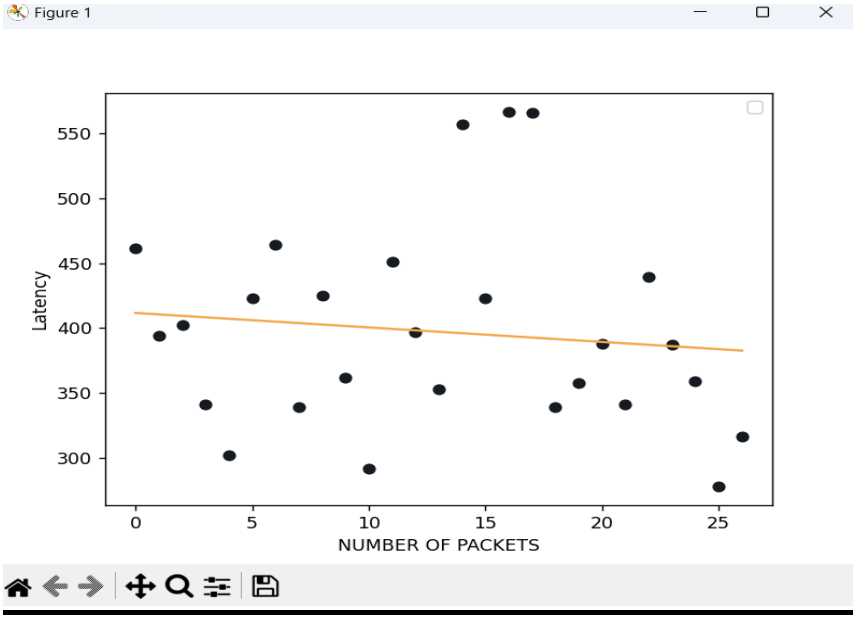
The LR function is called with the dataset path as an argument.

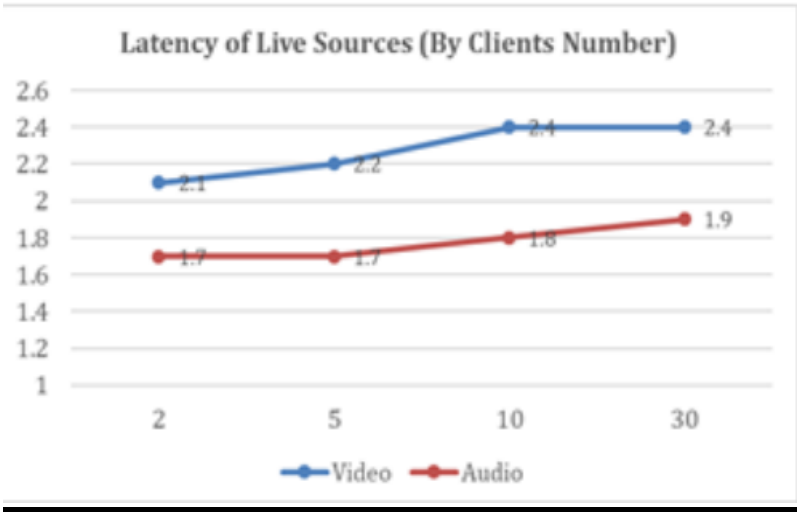
RESULT ANALYSIS

a) Comparison between existing work and our modification

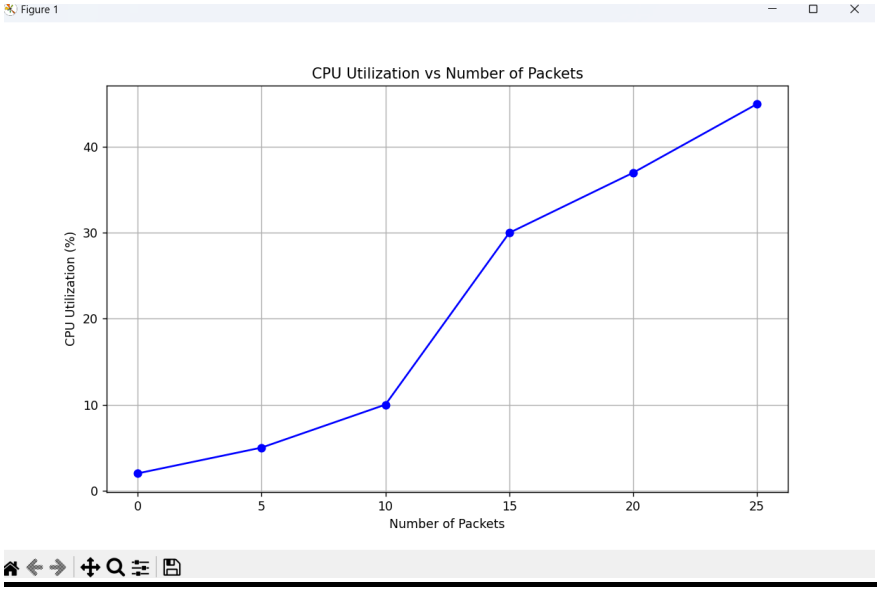
Metric	Base Paper (Existing)	Proposed Algorithm (Linear Regression)	Analysis
Latency	High latency.	reduced latency is achieved.	LR was successful in achieving the low latency aspect which helps in increased user immersion..
CPU Utilization	High CPU utilization.	Wise CPU utilisation.	LR effectively manages CPU resources reducing the strain on systems.
Algorithm Approach	NIL.	Implementation of Linear Regression	Linear Regression is a kind of Machine Learning technique that is used for optimization
Overall User Experience	Possibility of breakage of immersion because of lags in the video played.	Overall performance improvement with low latency and enhanced system response because of effective CPU utilization	Improved overall user experience by working on featured issues .

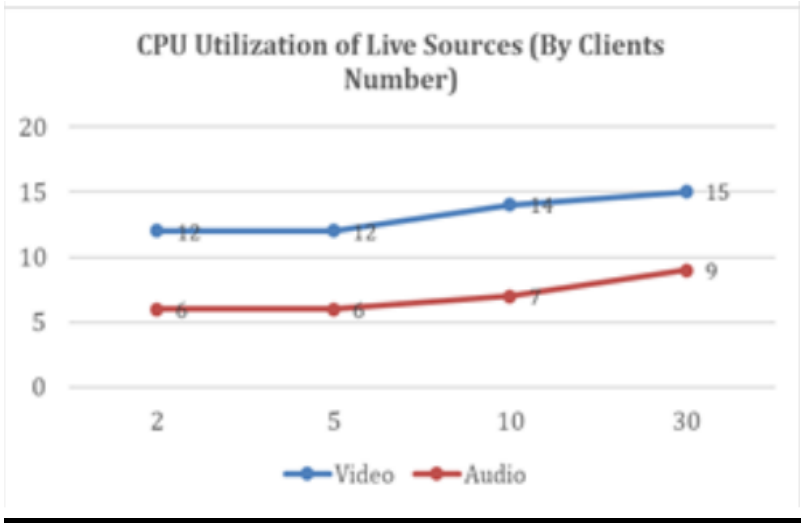
b)RESULT GRAPHS





Smart Latency Reduction: Our proposed algorithm uses machine learning techniques, specifically linear regression, which reduces the lag and delay in the video. This helps in ensuring the optimized timing of the data transmitted between client and server.





Efficient CPU Usage: Unlike fixed strategies employed by some existing approaches, our algorithm adapts in real-time. It intelligently utilizes the computer's resources, optimizing CPU power as needed for smooth streaming. This adaptability ensures that the system efficiently allocates resources, minimizing CPU utilization and preventing unnecessary strain on the system.

CONCLUSION:

a)Work done to remove the disadvantage:

Our proposed algorithm (Linear Regression) removes the problem of high latency and more CPU utilization

b)Input Parameter and Preprocessing techniques

To remove the existing problem, we have given the input parameters as sequence number, latency, and CPU utilization.

Sequence Number: acts like an identifier for the data in the dataset.

Latency: This is the variable that is used for the prediction of the expected result. Latency is the time delay between the initiation of an action and the occurrence of its effect,

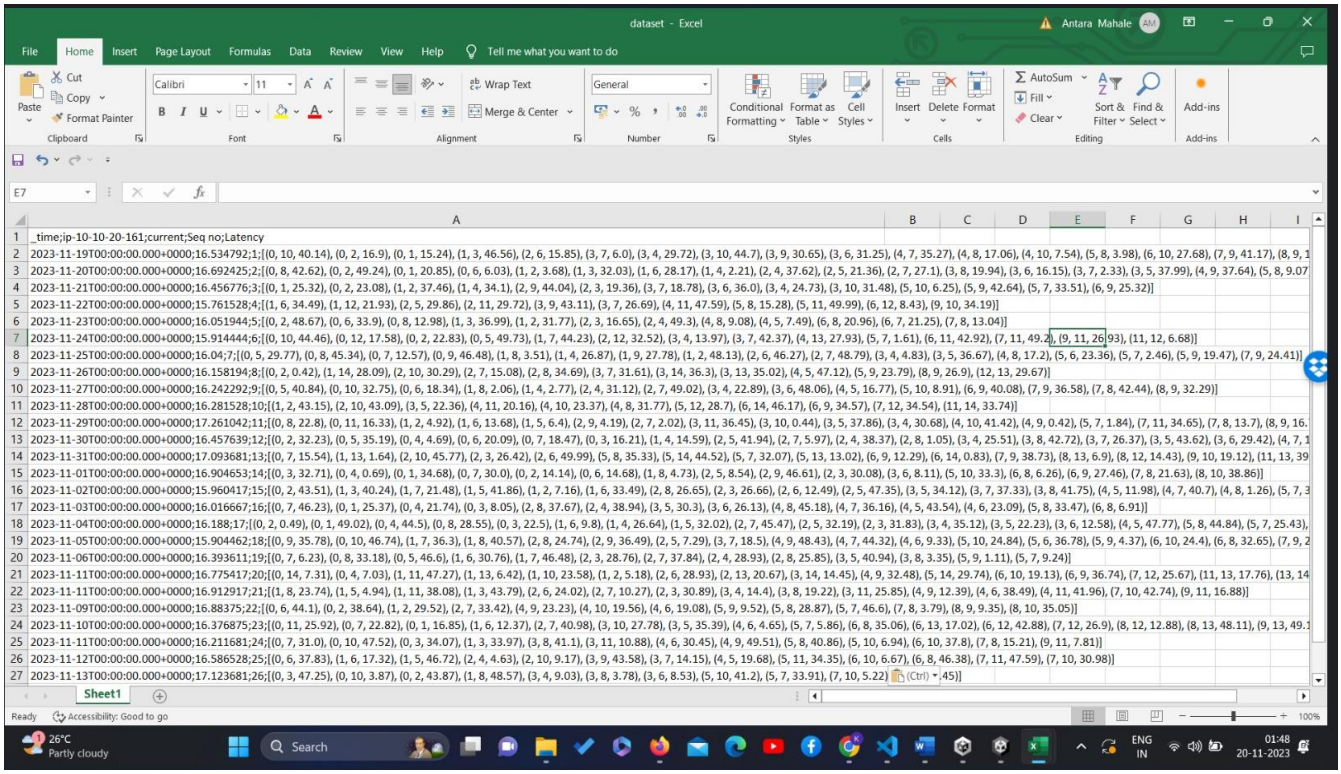
CPU utilization: It's considered as one of the features to predict latency. High CPU utilization is directly proportional to latency.

c)Algorithm impact:

It directly impacts the issue of low latency and ensures a lower CPU utilization while streaming the multimedia.

d)Data set and algorithms used:

Given below is the dataset used to compute the results, the algorithm used in our work is Linear Regression



e)Result:

The final results comprise working with Linear Regression which gives us the desired result of achieving low latency and lower CPU utilisation.

FUTURE SCOPE

Although the proposed advanced RTSP algorithm makes significant progress in addressing frame loss and improving the overall performance of real-time streaming protocols, there are several avenues for future research and development.

1.Integration of emerging technologies:

Consider integrating emerging technologies such as edge computing and 5G networks to increase the adaptability and performance of your algorithms in different network environments.

2.Optimization for devices with limited resources:

Explore techniques to optimize algorithms for resource-constrained devices and ensure efficient performance across a wide variety of platforms, including mobile and IoT devices.

3.Advanced machine learning model:

Research and implement more advanced machine learning models for error detection and recovery, aiming for more accurate prediction and reduced frame drops.

4.Dynamic QoS compatibility:

Extending the algorithm to include dynamic quality of service (QoS) adaptation enables real-time adjustments based on user preferences and network capabilities.

5.Robustness test in different scenarios:

We perform extensive robustness tests in various flow scenarios, including high-traffic periods, to evaluate the performance of our algorithms under difficult conditions.

6.User-centric compatibility:

Consider user-centric adaptation mechanisms that take into account user behavior and preferences and tailor the streaming experience to individual users.

7.Security considerations:

Perform comprehensive security analysis to identify and fix potential vulnerabilities in hardened algorithms to ensure a real-time streaming environment.

8.Standardization and industry acceptance:

Work for standardization, encourage the industry to adopt advanced RTSP algorithms, and work with relevant standards organizations to establish this algorithm as an accepted solution in the field.

SIMILARITY INDEX: