IT4010
Research Project

4th Year, 2nd Semester

Final report: IT21266232

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

Table of Contents

Table of Contents ii
1. Introduction 1
1.1 Background and Literature 1
1.2 Research Problem 1
2. Research Gap 1
3. Research Objectives 2
3.1 Functional Objectives 2
4. Methodology 3
4.1 Introduction to Methodology 3
4.2 System Design 3
4.3 Data Processing Pipeline 3
4.4 Compression and Encryption Implementation 4
5. System Architecture 5
5.1 Edge Computing on Raspberry Pi 5
5.2 Compression Modules 5
5.3 Encryption and Firebase Standards 5
6. Testing and Implementation 6
6.1 Testing Strategy 6
6.2 Performance Metrics 6
6.3 Implementation Environment 6
7. Results and Discussion 7
7.1 Data Compression Performance Results 7
7.2 Optimization of the Algorithm Excluding the Battery 7
7.3 Performance in Real Time 7
8. Conclusion 8
9. References 8

1. Introduction

1.1 Background and Literature

Autonomous patrol robots are transforming surveillance in environments such as warehouses, campuses, and public spaces with the potential to provide real-time monitoring and threat detection. The robots generate vast amounts of video and sensor data, which must be processed efficiently to maintain operational reliability with battery constraints. Edge computing on devices like the Raspberry Pi enables low-latency processing and reduces dependency on cloud infrastructure [15]. The LZ4 protocol, which is popular for its speed in compression, is optimum for processing videos in real time [1], while Zstandard offers improved compression ratios for sensor data [2]. AES-256 encryption ensures secure data transfer to cloud platforms like Firebase, whose scalability accommodates real-time data storage [7, 11]. Rule-based compression algorithms for battery-efficient compression have been put forward in IoT systems to make efficient use of energy [4]. This project integrates these technologies to develop an efficient, secure, and scalable patrol robot system.

1.2 Research Problem

Patrol robots face challenges in managing high-bandwidth video and sensor data on limited battery power. Cloud-based processing introduces latency and bandwidth issues, while uncompressed data transmission drains battery life and overwhelms network resources. The research problem is to develop an adaptive system that efficiently compresses video and sensor data on the edge using a Raspberry Pi, optimizes compression based on battery levels, and securely uploads data to Firebase for real-time monitoring and analysis.

2. Research Gap

Patrol robot systems typically utilize non-real-time battery limit or environmental state non-adaptive static compression methods. These utilize cloud computation with heavy latencies and lack efficient compression and secure cloud storage integration. The current research fills this gap by developing an amalgamated system with high-speed LZ4 video compression, Zstandard sensor data compression, AES-256 encryption, and Firebase integration, optimized based on rules for energy efficiency.

3. Research Objectives

3.1 Functional Objectives

The main goal is to construct a flexible patrol robot that optimates information processing, battery utilization, and secure data transfer. The system accomplishes the following goals:

3.1.1 Video Compression Using LZ4

LZ4 protocol will be applied for compressing strams of high resolution video. It will be done at a compression ratio of approximately 2:1 which would lower the required bandwidth by 50% without lower the quality of real-time surveillance.

3.1.2 Sensor Data Compression Using Zstandard

Compressing temperature, humidity, and motion data with the Zstandard protocol achieves up to 3:1 reduction with no loss of relevant information.

3.1.3 Battery-Aware Compression Optimization

A battery level monitoring algorithm will be implemented that adjusts data compression controls to sustain activity time for an additional 30%.

3.1.4 Secure Data Upload to Firebase

Sensor data and video fragments will be encrypted using AES-256 which guarantees security and will be uploaded in real-time to firebase for easy access any time.

3.1.5 Web-Based Administrator Dashboard

Create an alert-based manaul control mechanism on the web so users have full access to live video streams, control the sensors, check overall battery level, and manage robot navigation.

4. Methodology

4.1 Introduction to Methodology

The methodology is concerned with designing a patrol robot that performs video and sensor data processing with compression and upload using Raspberry Pi and Firebase. This encompasses system design, data processing, compression, encryption, and upload.

4.2 System Design

The system is structured as a modular architecture comprising of:

Data Acquisition: Gathers video streams and sensor data (temp, humidity, motion).

Compression Module: Uses Zstandard for sensor data and LZ4 for video.

Battery Conservation: Changes parameters with respect to battery levels, optimizing for battery life.

Encryption and Upload: Uses AES-256 for encryption and uploads data to Firebase.

Web Dashboard: Shows live data together with alerts for any critical conditions.

4.3 Data Processing Pipeline

The pipeline includes:

Data Collection: video 1080p at 30fps, and sensors at 1Hz sampling rate.

Compression: LZ4 for video and Zstandard for sensor data.

Battery Monitoring: Checking battery levels “on the go” (Must: >70%, Should: 30-70%, Critical:<30%).

Encryption: Compressed data is encrypted using AES-256.

Upload: upload the data to Firebase Realtime Database in a secure way.

4.4 Compression and Encryption Implementation

LZ4: set for aggressive compression speed to 500 MB/s.

5. System Architecture

5.1 Edge Computing on Raspberry Pi

The Raspberry Pi 4 (8GB RAM) serves as the edge computing platform, running all processing tasks:

- Video capture and compression.
- Sensor data acquisition and compression.
- Battery monitoring and optimization.
- Encryption and Firebase upload.

5.2 Compression Modules

- **LZ4 Module**: Processes video streams, reducing data size by 50%.
- **Zstandard Module**: Compresses sensor data, achieving a 3:1 ratio.

5.3 Encryption and Firebase Standards

- **AES-256 Encryption**: Encrypts compressed data with a 256-bit key.
- **Firebase**: Stores encrypted data in a Realtime Database, accessible via the web dashboard.

6. Testing and Implementation

6.1 Testing Strategy

Assessment was centered on:

- Compression Efficiency: Evaluating ratio and speed of compression.
- Battery Optimization: Assessing time of use prolongation.
- Data Security: Checking encryption strength and restrictive API access to Firebase.
- Real-Time Performance: Evaluating responsiveness and latency of the dashboard.

6.2 Performance Metrics

- Compression Ratio: LZ4 - 2:1 for video, Zstandard - 3:1 for sensor data.
- Processing Latency: compression and encryption below 100ms.
- Battery Life Extension: Compression up to 30% with adaptive algorithms.
- Upload Time to Firebase: Upload less than 200ms per data packets.

6.3 Implementation Environment

- Hardware: Raspberry Pi 4 with 4GB RAM and 32GB SD card.
- Software: Python 3.9 with Flask for the API, Firebase SDK, LZ4 and Zstandard libraries, cryptography libraries.
- OS: Raspberry Pi OS 64-bit.
- Development Tools: Postman for API testing and VS Code for programming.

7. Results and Discussion

7.1 Data Compression Performance Results

LZ4: Provided a compression ratio of 2:1 for the 1080p video while decreasing bandwidth by 50%, at a processing speed of 480 MB/s.

Zstandard: Provided a 3:1 compression ratio for the sensor data while keeping the information intact at a processing rate of 200MB/s.

7.2 Optimization of the Algorithm Excluding the Battery

The heuristic method decreased the operational limits modifier, such as lowering the video frame rate from 30fps to 15fps, leading to 28% enhanced performance duration during the test running over 6 hours.

7.3 Performance in Real Time

The system completed the processing of the video and sensor data in less than 100 milliseconds, while the Firebase upload was completed in less than 200 milliseconds. his led to the web dashboard refreshing with live data in under 50 milliseconds after receiving the infomation. The alerts too, were provided in real-time.

8. Conclusion

This paper presents an adaptive patrol robot system that efficiently compresses video and sensor data using LZ4 and Zstandard protocols, optimizes compression based on battery levels using a rule-based algorithm, and securely uploads data to Firebase. This system has enhanced energy efficiency, scalability, and security, making it suitable for real-world surveillance applications in dynamic environments.

9. References

[1] Y. Collet, “LZ4: Extremely fast compression algorithm,” GitHub Repository, 2011. [Online]. Available: https://github.com/lz4/lz4

[2] F. Drost, “Zstandard: A stronger compression algorithm,” Facebook Engineering Blog, 2015. [Online]. Available: https://engineering.fb.com/data/zstandard/

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, doi: 10.1016/j.future.2013.01.010.

[4] D. Mendez, I. Papapanagiotou, and B. Yang, “Internet of Things: Survey on security and privacy,” *IET Netw.*, vol. 6, no. 5, pp. 409–422, Sep. 2017, doi: 10.1049/iet-net.2017.0017.