Data Communication Project Proposal

Category

Team Introduction

> Team name : Berry Berry Raspberry

Team Member

2020320055 이성민

2020320112 김민준

2021320042 이한결

2021320056 장동윤

2021320087 이호준

2021320111 홍태선

2021320148 이상훈

2023320089 우규현

0. Topic

> Proposal for Data Communication Project : Building a Local Live Streaming System Using Raspberry Pi

1. Project Overview

The goal of this project is to design and implement a local live streaming system using a Raspberry Pi board equipped with V4L2C (Video for Linux 2 Control) packages. The system will capture video from a camera connected to the Raspberry Pi, process the video feed in real-time, and stream it over a local network. This project will leverage the V4L2C packages to enable efficient video handling and streaming capabilities.

2. Objectives

The primary objectives of this project are:

- To build a local live streaming system using the Raspberry Pi as a cost-effective platform for video capture and streaming.
- To utilize the V4L2C packages to interact with video devices, enabling real-time video capture and processing.
- To implement a streaming server on the Raspberry Pi that distributes the live video feed to devices connected to the same local network.
- To ensure the system operates with minimal latency and maintains a stable video quality suitable for real-time applications.

3. Technical Approach

The project will be divided into the following phases:

3.1. System Setup

- **Hardware Setup**: Configure the Raspberry Pi board, camera module, and network connectivity.
- **Software Installation**: Install the Raspberry Pi OS, necessary libraries, and packages, including V4L2C, FFmpeg, and VLC.
- **Network Configuration**: Set up the Raspberry Pi to function within the local network and assign a static IP address for stable communication.

3.2. Video Capture and Processing

- **V4L2C Configuration**: Configure V4L2C to interact with the camera module for video capture. This will involve setting the appropriate resolution, frame rate, and video encoding options.
- **Real-Time Processing**: Use FFmpeg or GStreamer to encode the captured video stream and prepare it for transmission.

3.3. Streaming Server Implementation

- **Streaming Protocol Selection**: Choose a suitable protocol (e.g., UDP) for streaming the video feed over the local network.
- **Server Configuration**: Set up a lightweight streaming server on the Raspberry Pi, such as VLC or a custom Python-based server using GStreamer.

3.4. Client Access and Playback

- **Client Devices**: Test video streaming on various client devices (e.g., laptops) connected to the same local network.
- **Analyze Data & Application :** With obtained streaming video, analyze the packet and implement a object tracker using YOLO.

4. Expected Outcomes

By the end of this project, the following outcomes are anticipated:

- A fully functional local live streaming system that can broadcast video captured by the Raspberry Pi camera over a local network.
- Documentation on the system configuration, including steps for setting up V4L2C and streaming services.
- Performance analysis report detailing the system's latency, video quality, and network bandwidth usage.

5. Conclusion

This project will demonstrate a practical application of data communication principles through the design of a local live streaming system using Raspberry Pi. It will provide a cost-effective and accessible solution for real-time video streaming, with potential use cases in surveillance, live events, and remote monitoring.

The proposed system not only showcases the capabilities of the Raspberry Pi in handling video communication tasks but also explores the use of V4L2C for efficient video device control. The outcome will contribute to practical knowledge in building streaming solutions and understanding the challenges of low-latency video transmission in local network.