VELAMMAL INSTITUTE OF TECHNOLOGY

Approved by AICTE - New Delhi
Affiliated to Anna University - Chennai
Accredited by NBA & NAAC

PUBLIC TRANSPORT OPTIMIZATION

TEAM NAME: Proj_224782_TEAM_2

Team Members:

NELAPATI GAYATHRI	(113321104066)
NIKSHITHA PRINCEE	(113321104067)
PARVATHAREDDY CHARMILA	(113321104070)
PASUPULETI MRUDHULA	(113321104071)

Phase 2 : Innovation

PROJECT OBJECTIVES:-

The objectives of the public transport is optimization should make the possibilities of efficiency, cost effectiveness, safety, and other important measures are as follows:

- **Enhance Efficiency and Reliability**: To ensure that the passengers should not waste there time by making the passengers to get information of the transport details.
- **Enhance Safety and Security**: To implement safety measures for the passengers and staffs.
- **Financial Sustainability**: To optimize the cost-effectiveness in the public transport services.
- **Data-Driven Decision-Making**: To collect and analyze data on passengers flows, route performance, and other measures to inform decision-making.

IoT SENSOR SETUP:-

Step1: Sensors selection

- **GPS Sensors:**

  Purpose: Tracking vehicle location and movement in real-time.

  Benefits: Helps in monitoring vehicle routes, on-time performance, and optimizing routes.

- **Passenger Counting Sensors:**

  Purpose: Monitoring passenger occupancy and load levels in vehicles.

  Benefits: Enables better route planning, resource allocation, and passenger safety.

- **Camera Systems:**

  Purpose: Video surveillance for passenger safety, incident monitoring, and driver behavior analysis.

  Benefits: Enhances security, aids in accident investigations, and ensures passenger safety.

- **Odometer and Speed Sensors:**

  Purpose: Measuring vehicle speed and distance traveled.

  Benefits: Supports route planning, fuel efficiency assessment, and adherence to speed limits.

- **Payment and Ticketing Sensors**:

  Purpose: Collecting fare and ticketing data.

  Benefits: Supports revenue management, ridership analysis, and fare optimization.

Step 2: Data Collection

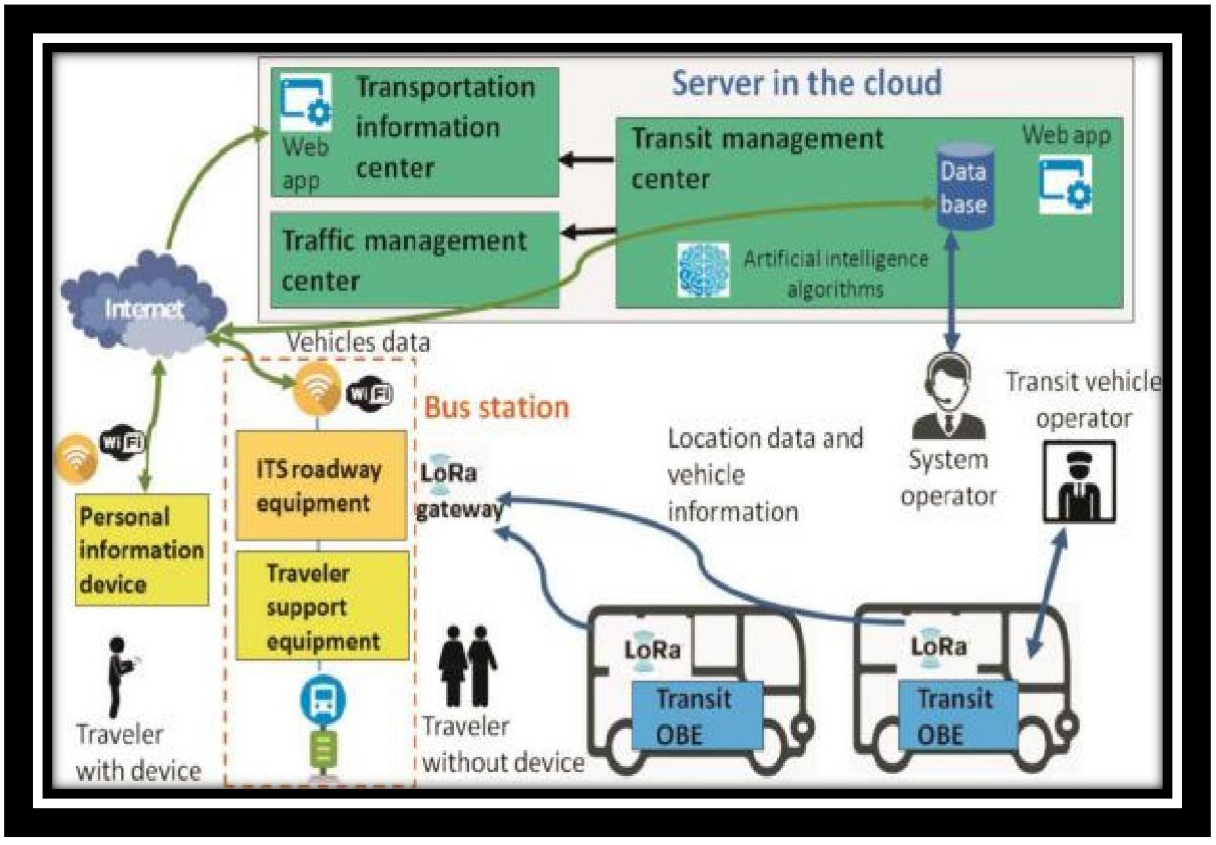
- **Automated Counting:** Install sensors or cameras in vehicles or at stations to automatically count passengers boarding and alighting. This data helps optimize bus/train frequencies.
- **Manual Surveys**: Conduct periodic manual surveys to gather detailed demographic and travel behavior data.
- **Traffic Patterns:** Integrate traffic flow data to anticipate congestion and optimize routes in real-time.
- **Weather Data:** Analyze weather forecasts to prepare for adverse conditions and adjust schedules/routes accordingly.

Step 3: Data Processing

- **Removing Noise:** Eliminate irrelevant or inaccurate data points from the dataset to ensure accuracy.
- **Handling Missing Data:** Implement techniques such as interpolation or data imputation to deal with missing data points.
- **Standardization:** Standardize data formats and units to ensure consistency across different sources.
- **Database Integration:** Store integrated data in a centralized database for easy access and analysis.
- **Optimization Algorithms:** Implement algorithms like linear programming or genetic algorithms to optimize routes, schedules, and resource allocation.

Step 4: Visualization and Control

- Public transport systems generate vast amounts of data, such as passenger counts, vehicle locations, and traffic conditions.
- Visualization tools can analyze historical data to optimize routes, taking into account factors like traffic patterns,peak hours, and passenger demand.



RASPBERRY PI INTERGRATION:

Raspberry Pi is a better choice for public transport optimization with IoT integration than Arduino.

Arduino is a microcontroller board designed for developing interactive electronic devices. It is relatively inexpensive and easy to use, making it a popular choice for beginners. However, Arduino has limited processing power and memory, which can make it difficult to run complex applications.

Raspberry Pi is a single-board computer that is more powerful and versatile than Arduino. It has a faster processor, more memory, and a wider range of connectivity options. This makes it a better choice for running complex applications, such as public transport optimization.

Additionally, Raspberry Pi has a larger community and more support resources than Arduino. This can be helpful if you need assistance with developing and deploying your public transport optimization application.

Here is a table that summarizes the key differences between Arduino and Raspberry Pi:

Feature	Arduino	Raspberry Pi
Processor	8-bit or 16-bit microcontroller	64-bit quad-core ARM processor
Memory	2-8 KB of RAM	4 GB or more of RAM
Storage	32 KB or more of flash memory	16 GB or more of microSD card storage
Connectivity	USB, serial, I2C, SPI	USB, HDMI, Ethernet, Wi-Fi, Bluetooth
Operating system	Bare metal or Arduino IDE	Linux
Community	Large	Larger
Support resources	Many	Many

Overall, Raspberry Pi is a better choice for public transport optimization with IoT integration because it is more powerful, versatile, and has a larger community and more support resources.

CODE IMPLEMENTATION:

To run the public transport optimization with IoT integration Python code, you will need to:

1. Install the required Python libraries:

- paho.mqtt.client
- scipy
- json

2. Replace the distance() function with a function that calculates the distance between two locations in your area. You can use a variety of methods to do this, such as the Google Maps Distance Matrix API or the OpenStreetMap Overpass API.
3. Start the MQTT broker.
4. Start the Python code.
5. Publish sensor data to the MQTT topic public_transportation.
6. Once all sensor data has been received, the code will solve the optimization problem and implement the optimized routes

```
import paho.mqtt.client as mqtt
import time
import json
from scipy.optimize import linprog
import math

# Define the MQTT broker address and topic
MQTT_BROKER_ADDRESS = "localhost"
MQTT_TOPIC = "public_transportation"

# Create an MQTT client
client = mqtt.Client()

# Connect to the MQTT broker
client.connect(MQTT_BROKER_ADDRESS)

# Subscribe to the MQTT topic
client.subscribe(MQTT_TOPIC)

# Define a callback function to handle incoming MQTT messages
def on_message(client, userdata, msg):
    # Decode the JSON message
    data = json.loads(msg.payload.decode())

    # Get the sensor data
    ridership = data["ridership"]
    location = data["location"]
    predicted_arrival_time = data["predicted_arrival_time"]

    # Update the public transport optimization model
    # ...

# Start the MQTT client loop
client.loop_forever()

# Define the public transport optimization model
def public_transport_optimization(ridership, locations, predicted_arrival_times):
    # Define the variables
    num_vehicles = len(locations)
    routes = [[] for i in range(num_vehicles)]

    # Define the objective function
    def objective(routes):
        return sum([sum([ridership[i] * distance(locations[i], locations[j]) for j in routes[i]]) for i in
range(num_vehicles)])
# Define the constraints
    constraints = []
    for i in range(num_vehicles):
        constraints.append(sum([ridership[j] for j in routes[i]]) <= 50)

    # Solve the optimization problem
    result = linprog(objective, constraints=constraints)

    # Return the optimal routes
    return routes
```

```
# Define the distance function
def distance(location1, location2):

  delta_lat = location2[0] - location1[0]
  delta_lon = location2[1] - location1[1]
  # Calculate the distance in kilometers.
  distance = math.sqrt(delta_lat**2 + delta_lon**2) * 111.325

  return distance
  # Calculate the distance between two locations
    # ...

# Implement the main function
def main():
    # Create a public transport optimization model
    model = public_transport_optimization()

    # Start listening for sensor data
    client.loop_forever()

    # Once all sensor data has been received, solve the optimization problem
    routes = model.solve()

    # Implement the optimized routes
    # ...

if __name__ == "_main_":
    main()
```

CONCLUSION:

In conclusion, public transport optimization is a multifaceted and dynamic process aimed at improving the efficiency, safety, and quality of public transportation systems. It involves a combination of planning, data collection, technological integration, and continuous improvement efforts. Key takeaways from public transport optimization include:

- **Efficiency:** Optimization efforts aim to make public transportation systems more efficient by improving route planning, scheduling, resource allocation, and maintenance practices. This leads to cost savings, reduced congestion, and improved passenger experiences.
- **Data-Driven Decision-Making:** Data collection and analysis play a central role in public transport optimization.Real-time and historical data help transportation authorities and operators make informed decisions and respond to changing conditions.
- **Passenger Satisfaction**: Enhancing the passenger experience is a core objective of public transport optimization. This includes improving on-time performance, reducing wait times, ensuring safety, and providing accessible services.
- **Environmental Impact**: Public transport optimization can contribute to reducing the environmental impact of transportation by promoting the use of clean energy sources and optimizing routes to minimize fuel consumption and emissions.
- **Safety and Security:** Safety and security measures are integrated into optimization efforts, with technologies like surveillance cameras, sensors, and real-time monitoring systems helping to ensure the well-being of passengers and staff.

In summary, public transport optimization is a complex and multifaceted endeavor that requires a combination of data-driven decision-making, technological innovation, and a commitment to meeting the needs of passengers and communities. As cities continue to grow and environmental concerns become more pressing, the importance of optimizing public transport systems is expected to increase, making transportation more efficient, accessible, and sustainable for all.