# CITY BUS MANAGEMENT SYSTEM

Saurabh Rai, Vishal Srivastava, Pranjal Sharma, Jishnu Ghosh

[1] *Under Graduate Student, Department of Computer Science and Engineering, SRM University, Chennai, India*
[2] *Under Graduate Student, Department of Computer Science and Engineering, SRM University, Chennai, India*
[3] *Under Graduate Student, Department of Computer Science and Engineering, SRM University, Chennai, India*
[4] *Under Graduate Student, Department of Computer Science and Engineering, SRM University, Chennai, India*

## ABSTRACT

*The people who use inner city public transportation vehicle want to get information about the different public transports form one stop to another, current status of the public transportation vehicles and they want to know the travel time of the vehicles both while travelling and waiting at the bus stops. This is an android application used to find out the public transportation vehicle details from one place to another place. User need to give the details of the source bus stop and destination stop. Accordingly it will display the details of the bus numbers which is going to that route. It will track the bus at the particular bus stop from which user wants to travel. It is a time saving application to the users. User can easily get information of the bus on that particular route. In this way user will be free of confusion about the buses. Through this developed system, moreover, it was ensured that the position and travel information of the vehicles through the GPS tracker installed inside the buses.*

**Keywords**: *Information System, GPS, Estimated Time of Arrival (ETA), Real-time System*

---

## 1 INTRODUCTION

The rapid development of cities and the increase in their population revealed the problem of transportation. The basic and solely solution for the transportation problem in the cities with big populations is popularizing the public transport systems. Although the public transport vehicles are used in the cities with big populations currently and the efforts to broaden its scope are executed. The complexity of public transportation system is the existence of more than one lines, more than one vehicles and routes causes we have difficulty in the managing system. There is a need for real time transportation information system for the public. The information thus, obtained by the passenger can aid in better travel planning and lower waiting time for the buses, thus using public transport effectively. With the increasing effectiveness of Intelligent Transport System solutions, the proposed prototype aims to provide real time estimates of arrival time, fleet management, etc. Thus connects the bus to the This prototype features a hardware implementation which acquires GPS data and computes the shortest path using Dijkstra's Algorithm, transmits the estimated time of arrival of the bus to the bus stops.

## 2 DESIGN

The system developed in this paper aims to minimize the waiting time to board the bus, manage and optimize the fleet of buses effectively and to provide a real time information of the arrival time of the bus. The inputs used are the latitude and longitude extracted from the GPS receiver of the bus. The extracted data from the GPS receiver is packed with additional information such as route number, registration number. The latitude and longitude is processed on the hardware mapped to compute the shortest path using Dijkstra's Algorithm on a predefined list of bus stops on the route. The location of the bus is also updated to a data logger maintained in a Google spreadsheet that acts as cloud storage. The data logger serves as the historic dataset to build the average time based ETA prediction model.

The outputs obtained after processing the data in real time are the ETA to the remaining stops and the communication of the position. Based on real time position updates, reliable information can be given to passengers and also better monitoring of the fleet of buses can be done.

## 2.1 System Architecture

The location of the bus is captured using a GPS receiver. The data from the GPS is streamed continuously and it comprises of the Latitude, Longitude, Number of Satellites that the GPS receiver is able to connect to, Deviation or Error in the position (in terms of meters),etc. The data is cleaned and the Latitude and Longitude is extracted from the data stream. The latitude and longitude are the inputs to the Dijkstra's Algorithm to compute the shortest distance to the destination (which is preset for the entire journey) from the current position. To facilitate the ability to record and store the movement of each bus, a data logger is designed. The data logger populates a Google spreadsheet and a CSV file with the timestamp, bus number, destination and location of the bus periodically. The use of Google Sheets is a cost-effective method and saves time setting up a custom cloud infrastructure. The purpose of the csv file is to bring in a redundancy to prevent data loss or data corruption and it acts like the black box in aircraft. Each bus is associated to a particular, unique topic (bus no.). This makes sure that a fast scalability of the prototype is possible. A prediction model based on the historic data is built to estimate the arrival time of the bus to any particular stop on the route. The bus can be monitored in real time by the administrator using a simple application. The passenger uses the proposed app to get real-time information with regard to the current location of the bus and an Estimated Arrival Time to his/her stop. The usage pattern of the proposed application is up to the user's discretion to get the necessary information that will be available with the app.

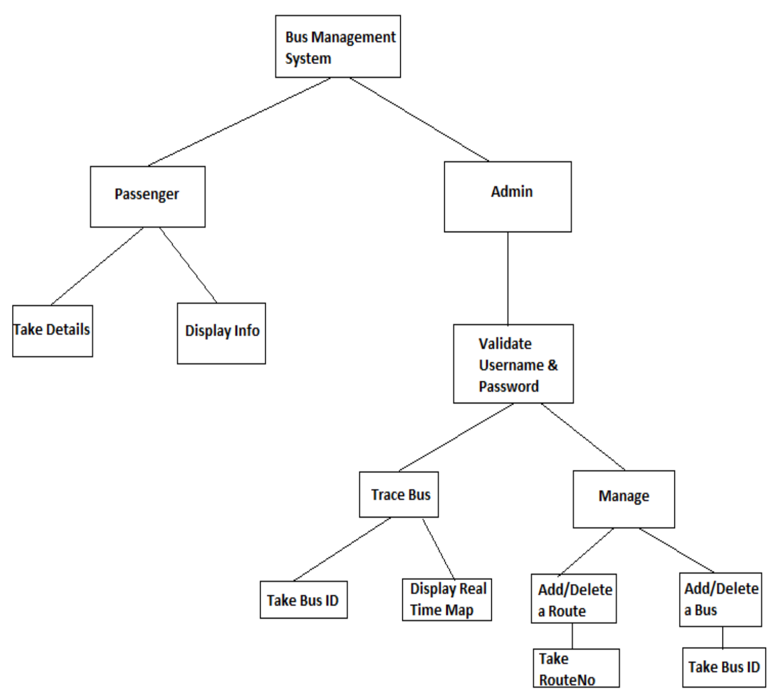


**Fig -1**: Architecture Diagram

### 2.2 Prototype Implementation

The prototype or in other words the hardware fitted onto each bus is an integration of Beaglebone Black and the Quectel L10 EVB Kit. This hardware runs a shell script that runs every time the Beaglebone Black boots up. The shell script triggers three other Python scripts. One of the scripts logs the position of the bus onto a Google Spreadsheet, transmits the location of the bus to the administrator and the end user via MQTT. The other script is responsible for running Dijkstra's Algorithm which is used to compute the shortest path to reach the destination based on the current latitude and longitude of the bus. The third Python script runs the prediction algorithm. A Multi-Layered Perceptron is used to compute the Estimated Time of Arrival (ETA) to the next stops on the route.

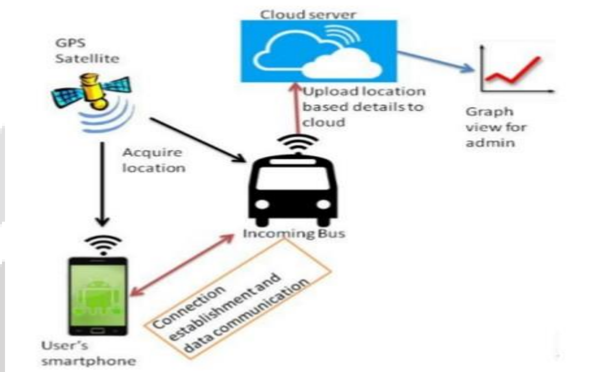


**Fig -2**: Overview of the System

### 2.3 Prototype of data on hardware

This section discusses on the sequence of steps that takes place on each bus to process and transmit its position. The order or the sequence of execution of the steps is identical to the control flow that was pictorially. There are two data streams that can be used to extract the position from the GPS module: GPGGA and GPGGL. Since, both are similar in giving out latitude and longitude, in this implementation we make use of the GPGGA stream to extract the latitude and longitude. It is to be noted that the waiting time defined in the algorithm as sleep(7000s) is subject to change for each device and its configurations. The connectivity to the MQTT broker is facilitated by means of 3G-USB Internet Dongle. The connection to the internet during the laboratory testing of the prototype was by providing Beaglebone Black a bridge from the host (Laptop). Once the GPS is in sync to the satellite constellation, the stream of live position data is continuous till the vehicle is powered down.

## 3 ADMIN CONSOLE

This section discusses the development and implementation of the Administrator's Console. The Administrator's Console is an android-application that is used to locate and track any bus in real-time by the administrator or the transport manager. This application is built using XML and Android Studio for the front-end display and styling of the web-page. A JavaScript code is embedded into the web page which does the backend processing. There are two JavaScript Application Program Interfaces (APIs) used in this implementation, namely: HiveMQTT-to connect to the MQTT broker and Google Maps API-to plot the position on an embedded map.

The administrator console has a very simple UI, The admin user selects the bus registration number first and then clicks on the "Connect" button. This establishes a connection between the MQTT broker and the information received from the MQTT broker is automatically updated in the respective form fields. The data that are updated automatically are the time, latitude and longitude. The latitude and longitude are also given as inputs to the Google Maps API which plots the position on the embedded map. This map is automatically refreshed as and when a new packet of data is received from the MQTT broker.

## 4 RESULTS AND ANAYTICS

This section builds the results and inferences as part of the pilot study. The system thus designed has been implemented in an university bus. The bus route comprises of 10 stops. The computation of ETA from historic data, comparison of the models used to estimate the arrival time is discussed in detail. The various inferences concluded from the analysis of the daily data include, traffic pattern and density, study on the driving pattern and also identification of possible bottlenecks in the route that affects ETA prediction.

### 4.1 Static model to estimate the arrival time based on average time

In this paper, the preliminary method to estimate the arrival time of the bus to any stop is computed with the historic data and average time. The average time in this context refers to the average time to travel between two consecutive bus stops along the route. Thus static models are used to compute the estimated time of arrival, based on the average time of both the start and end points. The model takes into account the actual time of departure from the bus stop with the ratio of the average times. It was also noted that this model gave inconsistent results, that is the mean and mean-squared error values that was computed for this model was higher than the average time for each stop model.

To validate the computed results and to develop a statistical model that was able to improve the prediction accuracy, a cumulative average model was developed. This model computes the cumulative average over the entire route for every trip instead of a running average between two consecutive bus stops.

The cumulative average model had an initial teething problem, with the prediction of ETA. But it was observed that over a trip with a significant delay, this model was close to predicting a close- enough ETA which was a fraction short of the actual arrival time. This model also led to the inference that the predicted time and the actual time at both the end-points of the trip coincided with the actual time, but the deviation in the estimated time for the stops in- between was very high. Thus, the cumulative model was not considered as a suitable model for a daily estimation of the arrival time. In the next section, we compare the three models developed on historic data and also infer a few important findings based on ETA prediction.

### 4.2 Comparison of various models used for analytics

In this section, the outputs from all the three static models are discussed. The comparisons of the predicted ETA from the different models are compiled together into a single spread- sheet. This spreadsheet is used to understand the performance of each model and also helps to modify the training and test data for the MLP, so that the anomalies can be adapted into the learning process. A better comparison of the models can be done by means of a graph which plots the deviation between the predicted and actual time against the bus stops for a specific interval of time. The graphs plotted shows the deviation between the predicted times from all the three models, the average time and the actual time at each stop. The deviation is measured in terms of seconds, as the deviations are significantly small. There are anomalies, which are ignored while plotting this graph. There are separate graphs that are plotted for each month for every trip made by the bus.

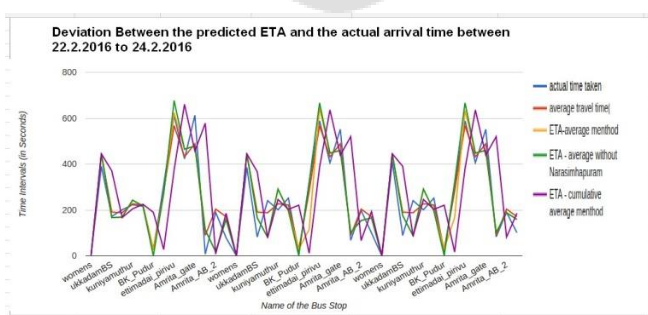


**Fig -5**: The comparison of predicted time from all the models

## 5 CONCLUSIONS

In this paper, a prototype to make buses communicate to the passengers in a City ecosystem has been proposed and developed. It is a tricky task, to plan a commute with the city public transportation. The passengers do not have the luxury of beforehand information about the fastest bus routes that ply to their destination, the bus frequencies are erratic making them less preferred. This situation might change with the introduction of a system that gives real-time and accurate information about the buses to the passengers. This setup comes with a fleet-management system that monitors and aids transport managers to improve efficiency of service, by diverting idle buses to on-demand routes in real-time. This system uses Internet of Things to build the ecosystem to connect the bus to the Internet and thus, to the passenger and transport managers. With Artificial Neural Networks and Statistical Modeling techniques, algorithms and models are built from historical data to predict the ETA efficiently.

This system is extremely efficient and is highly-scalable. In future the communication method between the buses can be done using IEEE 802.15P standard. The Dijkstra's Algorithm discussed in the earlier sections will work better when the buses are connected. Real time diversions to avoid traffic jams can be done thus increasing the productivity of the bus and reducing the delays for the passenger. In terms of the end-user the application can be further developed to suggest alternate routes based on lower waiting time or the shortest travelling distance, this can be done using the fusion of graph theory and machine learning algorithms. The application can be put to better use to the masses with language support for passengers to get the information in the regional language. This work can be extended to monitor the location of children, when they are travelling, and can also be used to remotely monitor the location of people in real-time.

## 6 REFERENCES

[1] Turnoff, Philip John, Darcy M. Bullock, Stanley E. Young, James Wasson, Nicholas Ganig, and James R. Sturtevant. "Continuing evolution of travel time data information collection and processing." In Transportation Research Board 88th Annual Meeting, no. 09-2030. 2009.

[2] Skiena, S. "Dijkstra's algorithm." Implementing Discrete Mathematics: Combinatorics and Graph heory with Mathematica, Reading, MA: Addison-Wesley (1990): 225-227.

[3] Hsieh, Han-Chuan, and Chi-Ha Lai. "Internet of things architecture based on integrated plc and 3g communication networks." In Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Con- ference on, pp. 853-856. IEEE, 2011.

[4] Tabish, Rohan, Abdulziz M. Ghaleb, Ramy Hussein, Farid Touati, Adel Ben Mnaouer, Lazhar Khriji, and Mohd Fadlee A. Rasid. "A 3G/WiFi- enabled 6LoWPAN-based U-healthcare system for ubiquitous real-time monitoring and data logging." In Biomedical Engineering (MECBME), 2014 Middle East Conference on, pp. 277-280. IEEE, 2014.

[5] Hunkeler, Urs, Hong Linh Truong, and Andy Stanford-Clark. "MQTT- SA publish/subscribe protocol for Wireless Sensor Networks." In Com- munication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on, pp. 791-798. IEEE, 2008.

[6] Thangavel, Dinesh, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. "Performance evaluation of MQTT and CoAP via a common middleware." In Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on, pp. 1-6. IEEE, 2014.

[7] Thiagarajan, Arvind, James Biagioni, Tomas Gerlich, and Jakob Eriks- son. "Cooperative transit tracking using smart-phones." In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, pp. 85-98. ACM, 2010.

[8] Williams, Billy M., and Lester A. Hoel. "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results." Journal of transportation engineering 129, no. 6 (2003): 664-672.

[9] Jeong R.H., "The Prediction of Bus Arrival time Using Automatic Vehicle Location Systems Data", A Ph.D. Dissertation at Texas AM University,2004.