Implementation of Real Time Bus Monitoring and Passenger Information System

Mrs.Swati Chandurkar, Sneha Mugade, Sanjana Sinha, Megharani Misal, Pooja Borekar

Computer Department, MAEER'S MIT AOE, Alandi (D) University of Pune 411015,Maharastra,India.

Abstract- In the daily operation of a bus system, the movement of vehicles is affected by uncertain conditions as the day progresses, such as traffic congestion, unexpected delays, and randomness in passenger demand, irregular vehicle dispatching times, and incidents. In a real-time setting, researchers have devoted significant effort to developing flexible control strategies, depending on the specific features of public transport systems. This paper focuses on the implementation of a Real Time Passenger Information (RTPI) system, by installing GPS devices on city buses. The Real Time Bus Monitoring and Passenger Information system is a standalone system designed to display the real-time location(s) of the buses in city. This research will enable the tracking devices to obtain GPS data of the bus locations, which it will then transfer it to centralized control unit and depict it by activating symbolic representation of buses in the approximate geographic positions on the route map. Specific software's will be used to interface the data received to the map. The main Objectives of this research work are :

1. RTPIS rolling display on bus stops – expected time of arrival in real time.
2. Web based interface for control room to monitor buses in real time.
3. Mobile application for end user to find out bus schedules and RTPIS.

Index Terms- GPRS, public transportation system, RTPIS, ETA, link updater.

I. Introduction

In the daily operation of public transport systems, mainly that of buses, the movement of vehicles is affected by different uncertain conditions as the day progresses, such as traffic congestion, unexpected delays, randomness in passenger demand, irregular vehicle-dispatching times, and incidents. Many passengers are often late to work, students are late for classes because they decide to wait for the bus instead of just simply using a alternate transportation. A variable message sign showing the showing the bus arrival time at bus stops could reduce the anxiety of passengers waiting for the bus. Disseminating arrival time information through other interfaces such as smart phone could make the public transit system more user-friendly and thus increase its competitiveness among various transportation modes. With the advent of GPS and the ubiquitous cellular network, real time vehicle tracking for better transport management has become possible. These technologies can be applied to public transport systems, especially buses, which are not able to adhere to predefined timetables due to reasons like traffic jams, breakdowns etc. The increased waiting time and the uncertainty in bus arrival make public transport system unattractive for passengers. A Real-Time Passenger Information System (RTPIS) uses a variety of technologies to track the locations of buses in real time and uses this information to generate predictions of bus arrivals at stops along the route [1]. When this information is disseminated to passengers by wired or wireless media, they can spend their time efficiently and reach the bus stop just before the bus arrives, or take alternate means of transport if the bus is delayed. They can even plan their journeys long before they actually undertake them. This will make the public transport system competitive and passenger- friendly. The use of private vehicles is reduced when more people use public transit vehicles, which in turn reduces traffic and pollution[1].

II. LITERATURE SURVEY

A considerable amount of money is spent on IT-based applications such as real-time, at-stop displays on public transport, but actual knowledge about the behavioral effects these have on customers or potential customers in real life is quite sparse. This paper presents a review of relevant literature, focusing specially on user response to public transport information via telephone, mobile devices, the Internet and at-stop displays. A number of studies have been initiated in the past to address the bus arrival time prediction problem.

Lin and Zeng [*2*] proposed a set of bus arrival time prediction algorithms for a transit traveler information system implemented in Blacksburg, Virginia. Four algorithms were introduced with different assumptions on input data and were shown to outperform several algorithms from the literature. Their algorithms, however, did not consider the effect of traffic congestion and dwell time at bus stations. Kidwell [3] presented an algorithm for predicting bus arrival times based on real-time vehicle location. The algorithm worked by dividing each route into zones and recording the time that each bus passed through each zone. Predictions were based on the most recent observation of a bus passing through each zone. However, this algorithm was not suitable for large cities where both travel time and dwell time could be subject to large variations. Generally speaking, these models are reliable only when the traffic pattern in the area of interest is relatively stable. One of their main limitations is that it requires an extensive set of historical data, which may not be available in practice, especially when the traffic pattern varies significantly over time [4].

III. ARCHITECTURE AND MODELLING

The main parts of RTPIS are application simulators, bus simulator and central data processing server. The architecture is shown in figure. These parts are briefly described in the subsequent sections

A. *Application simulator*

Pune Navigator has 3 applications as, the bus stop billboard display, the mobile application and the control room application. These services will request for the real time updates to the centralized server. The mobile application is android based application.

B. *Bus simulator*

The main functions of the bus simulator are as follows.

- To download names and coordinates of stops and points of interest from the server.
- To compute current location, direction.
- To transmit the computed information to the central server using GPRS.

It operates as follows – the GPS receiver in this unit computes the current location of the vehicle which is stored in bus simulator. The latitude, longitude of the buses is transmitted periodically to a central server using GPRS. The bus simulator unit initially downloads the names and coordinates of stops and POIs on the current route from the server. The Real time bus analyzer and computation contains the algorithm which calculates bus arrival time for each bus going through the corresponding route.

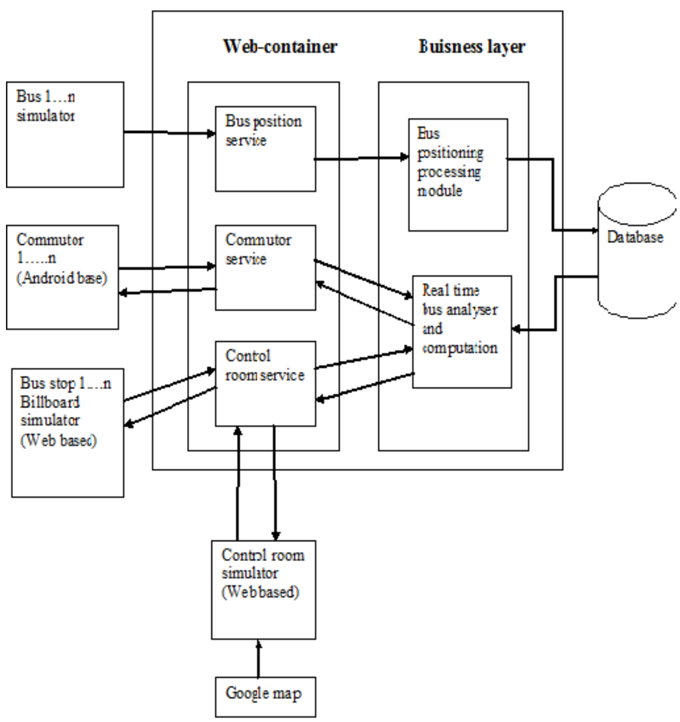


Fig.1 Architecture

C. *Server*

The server is at the center of RTPIS. The functions of the server are listed below.

- To maintain a database of all the routes, the buses that ply on a route, the stops along each route etc.
- To continuously receive location and speed from the vehicle units of all the buses.
- To calculate the ETA of all the buses at their next and subsequent bus stops.
- To reply to the android Google based queries requesting ETA of buses running between the the two specific stop from users; a GSM mode connected to the server transfers these queries to the server which processes them and reply the time.
- To host Internet web pages, which allow administrator to track buses in real time, see the route map of any route, and get the ETA for any route-stop Pair and plan trips from any source to any destination stop, at any time.

The server maintains a database of information pertaining to the buses, routes and stops in the form of tables. The server database can be organized in many ways, to reduce memory requirement, improve access speed, or reduce the number of queries. To improve the query speed, the tables related to buses are partitioned into static and dynamic ones. The Bus table stores static data while the bus position and log tables store dynamic data. The relation between the unique bus id, bus type (ordinary/luxury/…) and route number is stored in the Bus table. The position updates from the bus are stored in the Bus Position and the Bus Position Log tables. The direction is calculated in the vehicle unit by comparing time-separated position values with route details. The status of the bus changes to invalid, when its driver signals a breakdown. This helps the transport company to take suitable actions. The bus is excluded from ETA calculations based on this field. The Bus Position Log table stores a copy of the position update.

IV. IMPLEMENTATION

A. ALGORITHM

1. *Route creation*

A novel method has been developed to automate the process of creating new routes and populating the database, with little human intervention. To create a route, A bidirectional graph has been used. This graph will be used in ETA for calculating the estimated time of arrival. The bus stops will be represented as nodes and the route will be in the form of chain of links. A particular route will be identified by its unique id.

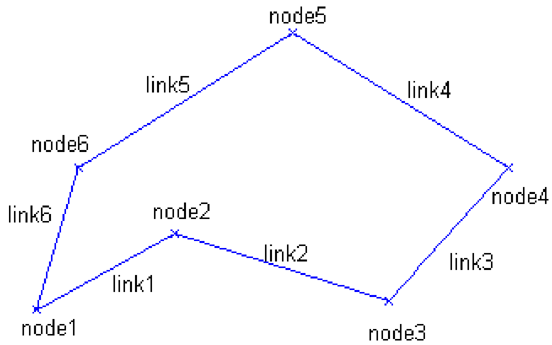


Fig.2 Route creation

2. ETA prediction

Arrival time prediction forms the core of any RTPIS system. The algorithm can be very simple, involving only a bus schedule table, zone based or could be very complicated, involving Artificial Neural Networks, space-time correlation and time series modeling. Bus schedule table and past location data can be used to predict arrival time.

This system provides a platform for executing any ETA algorithm, though here the implemented algorithm is simple one that adapts to changing traffic conditions. This algorithm works by recording the time it takes to traverse each link. Predictions are based on the present and past observations of a bus passing through each link. The past observations get lesser weight as time progresses; this reflects current traffic conditions better.

The predicted ETA at bus stops is bounded by an upper limit of one round trip time of the route, though the ETA can be predicted infinitely into the future by simply adding integral number of round trip times to the smallest ETA.

The ETA algorithm has two parts:

1. Link updater, which estimates the travel time for each link

2. ETA calculator, which calculates the ETA for every bus stop.

3. Link updater

Link updater calculates the link travel times required by the ETA calculator. Whenever a bus position update is received from the vehicle unit, the link updater calculates the travel times for all links traversed by the bus from the previous known position. The link updater requires distance of each link. The distance between two positions having latitude, longitude values is calculated by the following formula.

dlon = lon2 - lon1
dlat = lat2 - lat1
a = (sin(dlat/2))^2 + cos(lat1) * cos(lat2) * (sin(dlon/2))^2
c = 2 * atan2(sqrt(a), sqrt(1-a))
d = R * c (where R is the radius of the Earth)

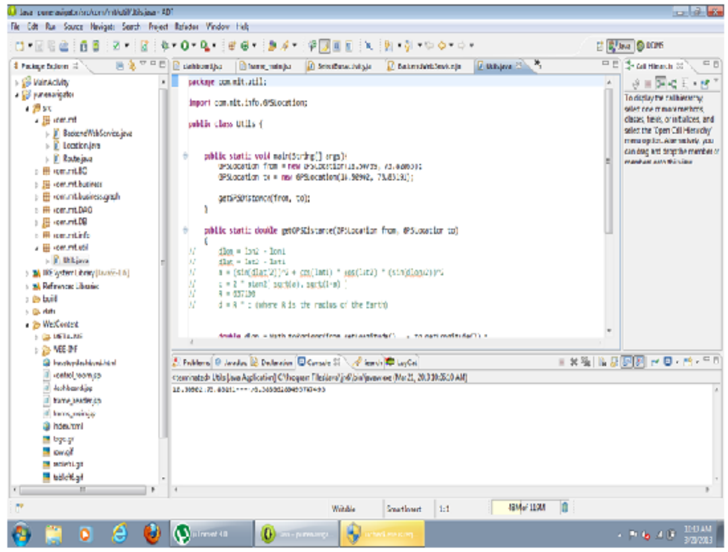
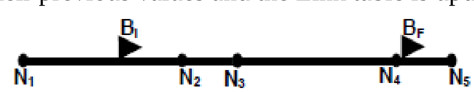


Fig. 3. Distance Calculation

The weighted average of the previous value and the actual travel time obtained for the current bus is stored as the link travel time in the Link table. For an update rate of two per minute used in trial runs, these weights give a good approximation of the average values, as well as track the recent trends. The link travel time is also common to all routes containing the link, so as to get the latest time estimate. This is the reason for sharing links between routes during route creation.

Link updater locates the bus position along the current route of the bus. The link updater then calculates the time required to reach the end of the current link and updates the estimated end time information in the Bus Position table. If the bus enters a new link, the entry time for the new link is stored in the Bus Position table against the bus and the travel time for all the crossed links is calculated. This time is also the exit time for the previous link. The time difference between the exit time and the previously recorded link entry time gives the link travel time for the crossed links. The travel times for links are a function of their lengths. Thus, when more than one link is traversed between updates, the individual link travel times are computed as *fractions* of the total travel time, with the fraction for link *i* being the ratio of the length of the *i*th link to the sum of lengths of traversed links. This makes sure that among the traversed links, shorter links have smaller travel times and longer links have larger travel times. The computed link travel times are averaged with their previous values and the Link table is updated.



1. Get bus id and route id
2. Get schedule id.
3. Provide this information to bus simulator.
4. update database.

4. ETA calculator

This program takes the current bus position, link travel times and estimated time to link-end to predict the ETA for all bus stops. ETA at a stop is the time taken for the nearest bus to reach the bus stop. It is calculated as the sum of travel times of the links, starting from the current bus position, up to the given bus stop.

B. SERVER UTILITIES

1. RTPIS at bus stop

The real time arrival information of buses at bus stops will be provided in the form of rolling displays. It will help the passengers to make efficient use of time. When this information is disseminated to passengers, they can spend their time efficiently and reach the bus stop just before the bus arrives, or take alternate means of transport if the bus is delayed [2]. This unit will periodically fetch the required ETA from the server via GPRS.

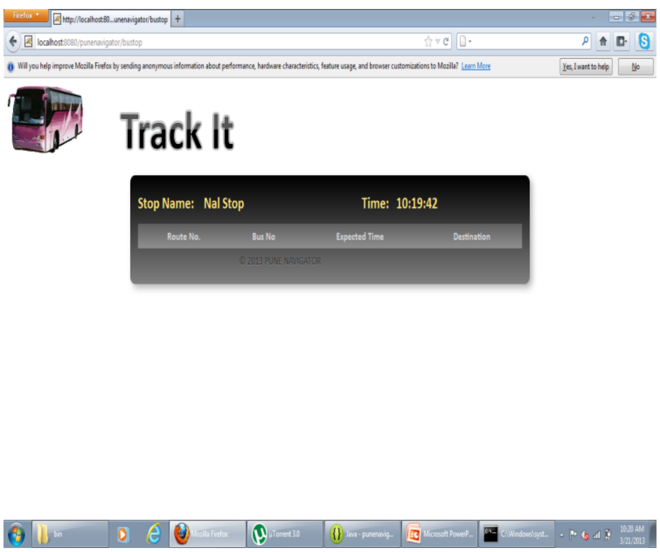


Fig. 4. Bus stop billboard

2. Smartphone Application

In today's world, everyone wants technology on fingertips. This system is providing a mobile application which will help the passengers to get bus arrivals at a particular stop. In this whole map of the city will be displayed and the passenger have to just give source and destination point. It will fetch the ETA of the requested route and provide the real time information to passenger. This will make the public transport system competitive and passenger- friendly.

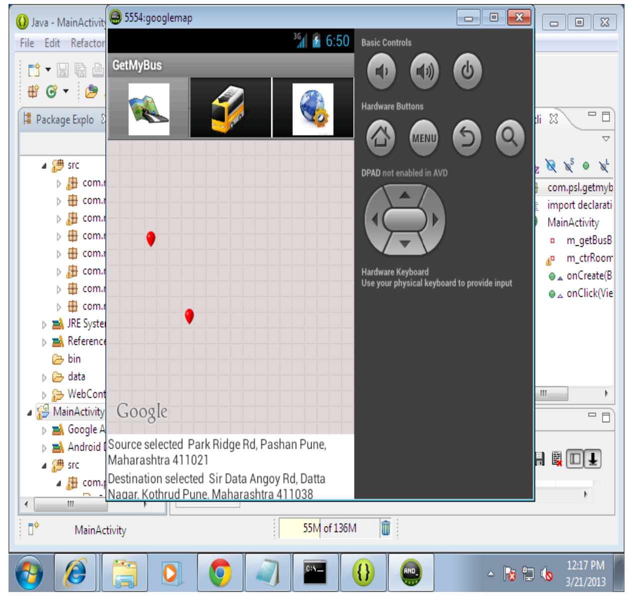


Fig. 5. Android Application

3. Tracking Buses

The whole map of the system will be provided to the administrator. Through this web page, user can view the present position of all the buses on the route map. This is done by getting the position of all the buses of a route from the database and then plotting it on the route map[5].

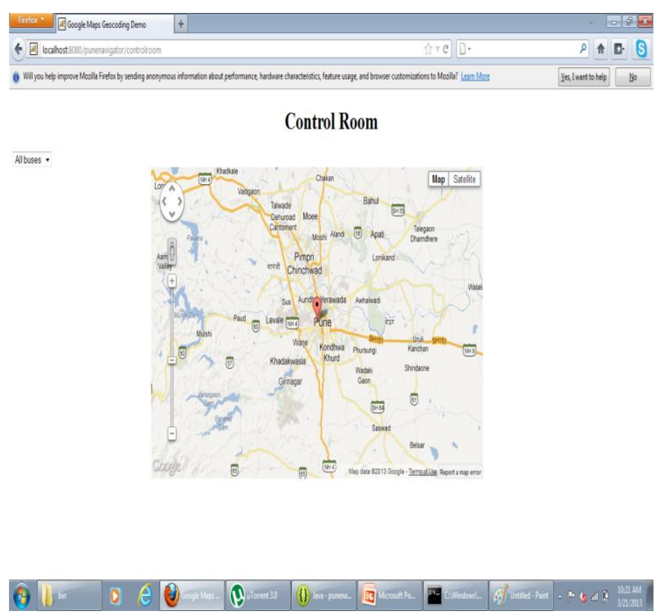


Fig. 6. Control room display

V. FUTURE SCOPE

As this system uses a combination of processing elements: PCs, Mobile Phones etc., there is a possibility of the overall system malfunction due to a particular type of attack, it is termed as Denial of Service (DoS) attack by malicious agents who might try to disrupt the function of the system. A Distributed Security

Scheme for Ad Hoc Networks can be used and to prevent this kind of attack. Such methodology will be studied to make this Real Time Passenger Information System more robust. A novel data hiding technique, based on Steganographic mechanism can also be used for security purposes. Here, the advantage lies in the fact that computationally costly encryption-decryption mechanism is avoided, thus making it suitable for a heterogeneous combination of processing elements, which are being used in present system. Here, many processing elements e.g. Mobile phone etc. lacks the processing power and battery power, which is required for traditional encryption-decryption system.

VI. CONCLUSION

In this paper, the partial implementation details of Real Time Bus Monitoring and Passenger Information System are stated. The RTPIS tracks the current location of all the buses and estimates their arrival time at different stops in their respective routes. Estimates are updated every time the bus sends an update. It distributes this information to passengers using display terminals at bus stops, web based GUI and smart phone application which is android based. This research serves the needs of passengers, vehicle drivers and administrators of the transport system.With the advent of GPS and the ubiquitous cellular network, real time vehicle tracking for better transport management has become possible.

REFERENCES

[1] Ganesh K and Thrivikraman M, Net Logic Semiconductors Pvt. Ltd., Banglore “Implementation of a real time passenger information system”.

[2] Lin, W.-H. and J. Zeng. “Experimental Study on Real-Time Bus Arrival Time Prediction with GPS Data”. In Transportation Research Record: Journal of the Tranportation Research Board, No. 1666, TRB, National Research Council, Washington, D.C., 1999, pp1019.

[3] Kidwell,B.“Predicting Transit Vehicle Arrival Times”. GeoGraphics Laboratory, Bridgewater State Colleg, Bridgewater, Mass., 2001.

[4] Wei-Hua Lin and Jian Zeng “An experimental study on real time bus arrival time prediction with gps data”.

[5] Seema S.R. and Sheela Alex, “Dynamic bus arrival time prediction using gps data”, 10th National Conference on Technological Trends (NCTT09) 6-7 Nov 2009.

[6] Dhaval Gada, Rajat Gogri, Punit Rathod, Zalak Dedhia, Nirali Mody, Sugata Sanyal and Ajith Abraham, "A Distributed Security Scheme for Ad Hoc Networks", ACM Crossroads, Special Issue on Computer Security. Volume 11, No. 1, September, 2004, pp. 1-17.

[7] Sandipan Dey, Ajith Abraham and Sugata Sanyal "An LSB Data Hiding Technique Using Prime Numbers", Third International Symposium on Information Assurance and Security, August 29-31, 2007, Manchester, United Kingdom, IEEE Computer Society press, USA, ISBN 0-7695-2876-7, pp. 101-106, 2007.

[8] Yu-Shuen Wang and Ming-Te Chi, “Focus+Context Metro Maps”, IEEE transactions on visualization and computer graphics, vol. 17, no. 12, december 2011.

[9] A. Guin, “Travel Time Prediction Using a Seasonal Autoregressive Integrated Moving Average Time Series Model”, IEEE ITSC, 2006.

[10] Taehyung Park, Sangkeon Lee, Young-Jun Moon, “Real Time Estimation of Bus Arrival Time under Mobile Environment”, ICCSA, 2004.

AUTHORS

First Author – Mrs.Swati Chandurkar, Computer Department, MAEER’S MIT AOE, Alandi (D), University of Pune 411015, Maharastra, India, Email: swati.bhutekar@gmail.com

Second Author – Sneha Mugade, Computer Department, MAEER’S MIT AOE, Alandi (D), University of Pune 411015, Maharastra, India, Email: sneha.mugade99@gmail.com

Third Author – Sanjana Sinha, Computer Department, MAEER’S MIT AOE, Alandi (D), University of Pune 411015, Maharastra, India, Email: sanjanasinha.sinha@gmail.com

Fourth Author – Megharani Misal, Computer Department, MAEER’S MIT AOE, Alandi (D), University of Pune 411015, Maharastra, India, Email: meghamisal121@gmail.com

Fifth Author – Pooja Borekar, Computer Department, MAEER’S MIT AOE, Alandi (D), University of Pune 411015, Maharastra, India, Email: pooja.patil030@gmail.com