IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

Bus Bunching Avoidance System

Mrityunjaya D H[1], Pankaj Walvekar[2], Manoj Awaji[3]

Student, Electronics and Communications Department, BVBCET, Hubli, India [1, 2, 3]

Abstract: In the era of ever increasing demand for vehicles brings forth various inherent problems with the large vehicle populous such as traffic congestion, fatal accidents, vehicle bunching etc. One of the problems caused by buses specifically is bus bunching. It is the tendency of buses to converge at a bus stop which is majorly due to traffic conditions, where buses tend to clump. Another reason is due to communication gap between buses drivers due to which multiple buses converge at the same bus stop. In this paper an idea to overcome this problem is presented.

Keywords: Microcontrollers, RF signals, GSM, Protocols, Antennas.

I. INTRODUCTION

Bus bunching is one of the foremost problems in bus traffic which causes a large fuel wastage and long waiting time for passengers at the bus stop.

Consider a journey of two buses on a busy road. A bus that is only slightly late to the bus stop will, in addition to its normal load, pick up passengers who would have taken the next bus had the first bus not been late. These extra passengers delay the first bus even further. In contrast the bus behind the late bus has a lighter passenger load than it otherwise would have, and may therefore run ahead of schedule. A late bus tends to get slower in its approach as it completes its run, while the bus following it tends to get closer to that bus. Eventually these buses form a pair, one right after another, and the service deteriorates as the headway degrades from its nominal value. The buses that are stuck together are called a bus bunch or banana bus; this may also involve more than two buses. Another of the cause being communication gap between the bus drivers as no medium is provided for contacting the other bus driver. Also traffic jam on busy roads tends to clump the buses at a single location even though they were far apart in the first place. These effects can be theorized to be the primary causes of bus bunching and hence making the bus transportation less reliable.

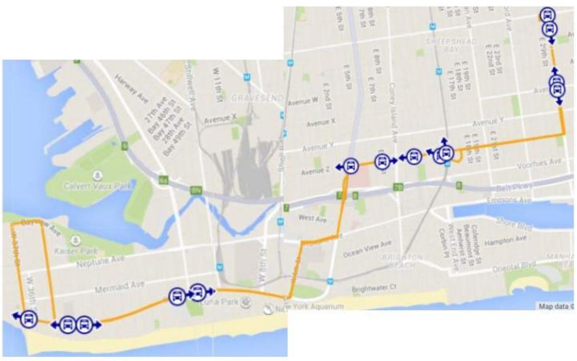

Fig.1 Map where blue dots resembles bus

Consider fig.1 where there are 2 instances of banana bus (bunched buses) in the map. In this scenario only one of the buses is enough to serve the passengers of the bus stop at the instant and the other bus which waits in the queue at the bus stop just burns fuel without serving its purpose. Even if one of the buses is removed from the scenario it won't affect the normal flow of transportation. This is a case which demonstrates how bus bunching leads to fuel wastage which can be otherwise saved if bus bunching is averted.

II. LITERATURE SURVEY

According to a survey carried out on the distance travelled by buses at the peak hours of day, Fig.2 is a plot devised to visualize the deviation of distance (in kilometres) from the scheduled/ideal value to the actual value.

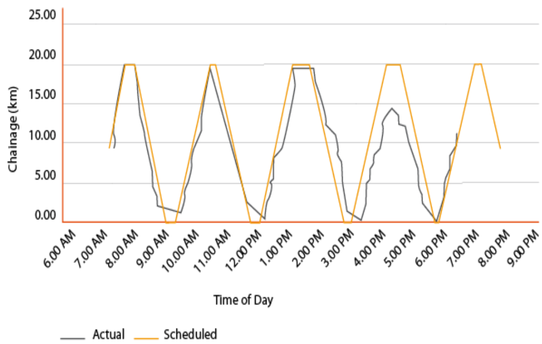

Fig.2 Graph of Distance covered (km) Vs Time

The actual value is in tandem with the scheduled value in the morning hours as a consequence of lower traffic conditions. But during the peak hours of the day (12 p.m. to 6 p.m.) we see a greater deviation from the scheduled value which is a result of traffic conditions and them being traffic jams, bus bunching etc [7]. So if either of the problems is solved to certain extent, it will contribute towards approximating the plot towards the scheduled value and help in effective utilization of the bus transportation services across the city at any time of the day [5].

III. PROPOSED IDEA

The principal problems that need to be attended to solve bus bunching problem are communication gap between drivers, bus clumping as a result of traffic conditions. Now

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

to give a brief description of the methodology used to solve these problems. We begin with installing GPS module inside every bus operating in the city so as to get a hold of the buses location at every instant of time. The need of location of the buses is established on the need of distance between the buses travelling one before another. After two buses are located on map, the distance between them can be calculated using the co-ordinates provided by the GPS module [9].

IV. CONCEPTUAL DESIGN

To calculate the distance, a certain system must be provided with both the co-ordinates. The system needed for calculation is installed inside every bus which majorly comprises of a microcontroller unit. Each bus stores its own location inside its own GPS module and this demands for a way to share this data between the buses travelling one behind the another which in turn demands a connection to be set up between these buses so that they can transfer the data between them [4]. To do so, a RF (Radio frequency) module is installed in a bus which tracks buses in a radius of 1-2 km (i.e. tracking a bus in its vicinity so as to share data and establish communication between the drivers).

Each RF signal is made unique for an individual bus so that two buses out of many hundreds of buses travelling across the city can recognize each other and become eligible for data exchange. After the recognition, they share the number (ID) of their GSM modules so that the data can be sent. Once the data is exchanged, the microcontroller unit which centrally coordinates all of the procedure explained above calculates the distance between the buses using the shared co-ordinates. There is a possibility that multiple buses may be present in the vicinity of the bus, only the bus closest to it is considered for data sharing which is achieved via an algorithm designed to choose the proper bus [2]. The microcontroller units of both the buses contemplating the distance between buses derive a speed range which if followed by the buses regularly will maintain a proper distance between them. Conversely, if one of the drivers decides not to follow the speed range a counter speed range is relayed to the other driver so that the distance is maintained. In case a bus halts at a bus stop, zero speed of that bus can be treated as a special event and that speed must be sent to the other bus. The other microcontroller recognizes this event and intimates the driver to skip the next bus stop as it is already being catered by another bus.

The necessary data is displayed on LCD screen to the driver. By this method, communication is established between the drivers which is primordial in addressing the bus bunching issue. An alternative to this solution can be designing a central system which is connected to every bus and relays the necessary data dynamically. But frequent connectivity disruptions and local noise interference make this type of system less reliable. Also sudden failure of the central system can lead to bus handling chaos and which may cause big problems than just bus bunching. Also the application doesn't require the need of a central system because a simple microcontroller along with some modules is enough to serve the purpose.

V. DETAILED DESIGN

Fig.3 shown is a transparent box representation of system installed in the bus to establish communication between buses. It encloses the necessary modules and displays the flow of information between them.

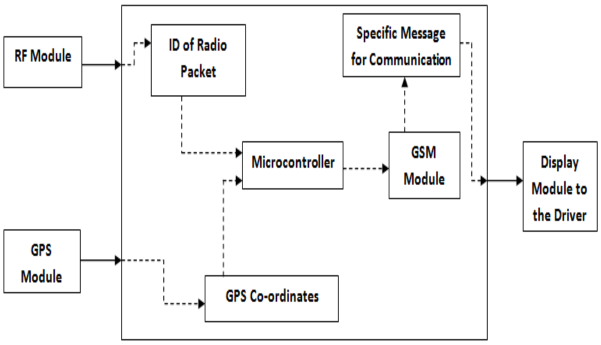

Fig.3 Data flow and module representation

The modules comprising the system are:

A. RF Module

It is the module necessary to track buses in its vicinity and establish connectivity with the nearest bus. Also each module should generate a unique RF signal so that each module has its own unique identity allowing easy identification of a single bus among the many buses operating in the city which in turns helps in connecting the two specific buses which are close to each other and finally allowing data sharing. The module under consideration is WIR-1186 which is a trans-receiver (SOC) with an MCU for wireless network control and hardware interface which supports point to multipoint serial communication over the air and range of 1-2 km. It is capable of transmitting RF signals necessary to detect buses and receiving them so as to establish connection [1]. Consider Fig.4, where RF module is divided into two sub modules which are: i. Radio module: It is responsible for transmission and reception of data wirelessly via an antenna. Maximum data rate is around 22.5 kbps and frequency options available are between 865-869 kHz.

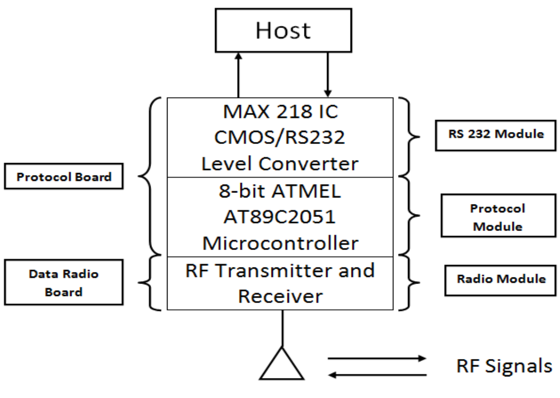

Fig. 4 Block representation of RF module

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

ii. Protocol module: It is responsible for data processing i.e. in simple words making an RF signal unique. Here a microcontroller is installed in the module which is responsible for signal processing and RS232 connects this module serially to the central microcontroller (host). Now coming to the part where RF should be encrypted to make it unique, a protocol is followed by the firmware on this module to encrypt the signal. This protocol is termed as RFM link layer protocol [12]. Fig.5 is format of radio packet by RFM link protocol. The parts that are of interest in our application are:

1. Packet number: It is of size of 8 bytes which is unique to one RF signal packet.
2. Message data: It is of size of 32 bytes which send the required message, be the number (ID) of GSM module or speed of vehicle in a special event (as explained under section II).

Start Symbol(0x55)
To/From
Packet Number
Data Size
Message Data
16-Bit FCS
Radio Packet

Fig.5. Format of a Radio packet.

B. GPS Module
This module is used to find the location of the bus on map by retrieving its co-ordinates. The module used for this purpose is SKM58. It is based on NMEA protocol for location tracking [11].

Table 1 NMEA protocol

Filed	Length (Bytes)	Description
$	1	Each NMEA message starts with a '$'.
Talker ID	1~2	'GP' for a GPS receiver.
NMEA message ID	3	NMEA message ID
Data Field	Variable, depend on the NMEA message type	Data Field, delimited by comma ','.
*	1	End character of data field.
Checksum	2	A hexadecimal number calculated by exclusive OR of all characters between '$' and '*'.
<CR><LF>	2	Each NMEA message ends with 'CR' and 'LF'.

Table 1 is the standard format of a NMEA message which is received by the GPS module. The positional data is stored in the output message under GPRMC in the NMEA protocol, where RMC is recommended minimum position data such as position, velocity and time (see Table 2).

The RMC message stores all the necessary data required for tracking the buses location i.e. latitude, longitude, N/S, E/W etc [10].

Table2. GPRMC

$	Each NMEA starts with a '$'
GPRMC	Message ID
UTC Time	Time in Format ' hhmmss.sss '
Data Valid	' V ' = Valid 'A' = Invalid
Latitude	Latitude in format 'ddmm.mmmm' (degree and minute)
N/S	' N ' = North ' S ' = South
Longitude	Longitude in format 'dddmm.mmmm '(degree and minute)
E/W	' E ' = East ' W ' = West
Speed	Speed over Ground in knots
COG	Course over Ground in degrees
Date	Date in format ' ddmmyyyy '
Magnetic Variation	Magnetic variation in degree , not being output
E/W	Magnetic variation E/W indicator , not being output
Positioning mode	' N ' = No fix ' A ' = Autonomous GNSS fix ' D ' = Differential GNSS fix
*	End Character of data field
Checksum	Hexadecimal Checksum
<CR><LF>	Each of Sum

C. GSM module
It is necessary for transfer of messages between buses when triggered by the microcontroller. The module used is SIM300 which is a tri band GPRS engine [6].

D. Microcontroller:
It centrally co-ordinates all the modules present in the system. To do so, all the modules must be interfaced with it. The microcontroller used for this purpose is AT89S52 (which is a 8 bit microcontroller with onboard watchdog timer).
Fig.6 is a circuit for interfacing of GSM module with the chosen microcontroller.
Fig.7 demonstrates interfacing of GPS module with the chosen microcontroller.
Fig.8 demonstrates interfacing of GSM with AT89S52.
Entertaining the possibility that multiple buses may be present in buses vicinity, an algorithm must be developed for the microcontroller to find the bus closest to it.

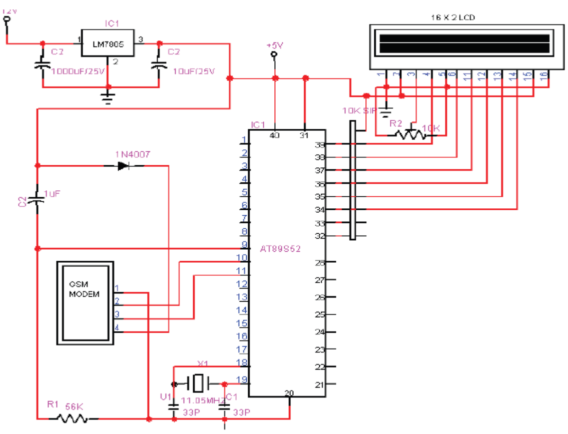

Fig.6. Interfacing GSM with AT89S52

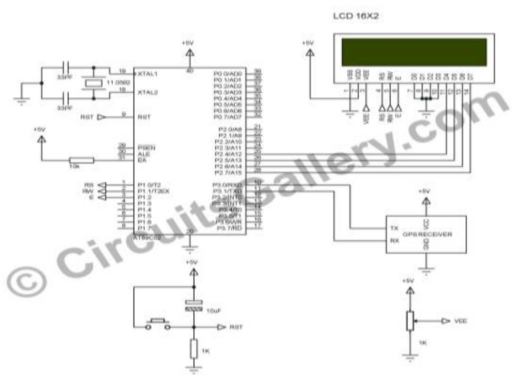

Fig.7 Interfacing AT89S52 with GPS module

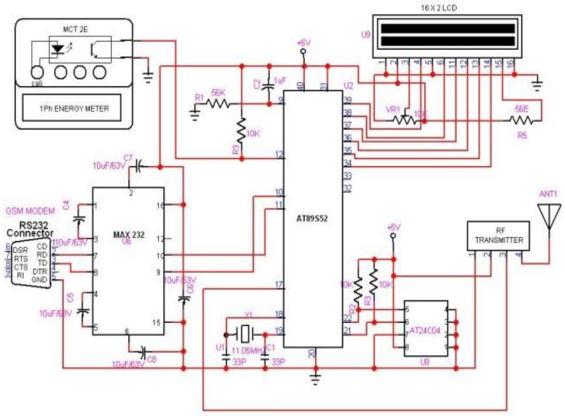

Fig.8 Interfacing AT89S52 with RF module

The algorithm is explained in the following steps:
1. First retrieve the location co-ordinates of all he buses in the vicinity.
2. Then calculate the distance between the current bus and the remaining buses individually using the co-ordinates.
3. Find the least distance amongst the lot which gives the nearest bus.
4. Establish connectivity and then share data via GSM module.
These interface circuits serve as a proof of concept that the required modules can be interfaced with the chosen microcontroller.

VI. INNOVATIVE APPROACH

Some of the existing solutions available for the bus bunching problem are,

1. Synchronization of bus time tabling in which departure time of each bus trip is determined which is under the domain bus network strategic planning. Such a system was practiced and studied in Monterrey, Mexico.
2. Kinetic clustering and jamming transitions in a car-following model for bus route: It is based on an older car following model which is extended to the bus mimic behaviour by studying the phase transitions and bunching in the bus route model by both simulation and linear stability analysis.

The models discussed above rely on statistical data and they do not take into consideration the real time constraints that are imposed by traffic conditions.
On the other hand, our system depends on the real time location of the bus making it dynamic also more reliable as decisions such as the proposed speed range are taken instantaneously [8].

VII. CONCLUSION

This paper presented the causes of bus bunching one of the major problems troubling the bus transportation services and explained the impact of the problem using statistical analysis. It explained about a proposal to tackle the problem and explained various standards and protocols that are pivotal to the success of the solution. Tools such as transparent box were employed to explain the working and interconnectivity of modules present in the solution. The proposed solution was compared with the existing solutions for the problem and was found to be innovative in terms of reliability, accuracy and also establishing a communication between buses.

REFERENCES

[1] A. Adamski. Flexible Dispatching Control Tools in Public Transport. Advanced Methods in Transportation Analysis, pages 481–506, 1996.
[2] P. Chandrasekar and H.C. Chin. Simulation evaluation of route-based control of bus operations. Journal of Transportation Engineering, 128:519, 2002.
[3] M. Dessouky, R. Hall, L. Zhang, and A. Singh. Real-time control of buses for schedule coordination at a terminal. Transportation Research Part A, 37(2):145–164, 2003.
[4] A.J. Finnamore and R.L. Jackson. Bus control systems: their application and justification. Crowthorne [Eng.] : Transport and Road Research Laboratory, 1978.
[5] Jerry Jariasunant, Daniel B. Work, Branko Kerkez, Raja Sengupta, Stephen Glaser, and Alexandre Bayen. Mobile Transit Trip Planning with Real-Time Data. accepted for TRB presentation, forthcoming.
[6] JK Jolliffe and TP Hutchinson. A behavioural explanation of the association between bus and passenger arrivals at a bus stop. Transportation Science, 9(3):248, 1975.
[7] M.D. Hickman and N.H.M. Wilson. Passenger travel time and path choice implications of real–time transit information. Transportation Research Part C, 3(4):211–226, 1995.
[8] X.J. Eberlein, N.H.M. Wilson, and D. Bernstein. The holding problem with real-time information available. Transportation science, 35(1):1, 2001.
[9] http://www.8051projects.info/resources/circuitdiagramforinterfacing sim300gsmmodemtoat89s52.15/
[10] http://blog.sumitbhattarai.com.np/2013/09/gpsinterfacingwithat89s52.html
[11]`https://www.pantechsolutions.net/projectkits/interfacinggpsl10with8051
[12] http://robokits.co.in/wirelesssolutions/wirelessrfseriallink865869mhz2kmrange