Empowering Bus Transportation System Using Wireless Sensor Networks

Ankit Kesharwani [†*], Vaishali Sadaphal[‡], Maitreya Natu[‡]
[*]Student author
[†]Indian Institute of Technology, Kanpur, India
[‡]Tata Research Development and Design Centre, Pune, India

***Abstract*—A proper and systematic public transport plays a major role in economic development and well being of the people in any country. But public transport in most of the developing countries is not running properly because of the lack of systematic planning and monitoring. In this paper, we have used WSN for monitoring operations of bus transportation systems. We propose use of minimal information viz. record of arrival time of the buses at the bus-stops, for improvement of bus transportation system. We present algorithms to perform various analysis operations namely detection of bus delays, likely cause of the delay and prediction of likely arrival time of buses at the bus-stops. We also present and compare various design choices of WSN available for bus transportation system.**

I. Introduction

Public Transport is one of the important infrastructures of any country. In developing countries like India bus transport consists of 90% of public transport [1]. However, lack of systematic mechanism to monitor and manage the bus-network is leading to lack of predictability of the bus network. It gets difficult to identify causes behind delays, or predict the arrival times. Bus transport system faces the ever increasing problem of traffic and congestion. Without a well-deployed monitoring system, it becomes very difficult to plan for optimization and growth.

The operations of bus-transport systems can be significantly improved by monitoring the bus operations and analyzing them to providing useful information both to the travellers and bus operating authorities. Below we list some examples:

- Travellers can benefit from information such as current bus location, expected time of arrival of a bus at a bus-stop, availability of seating space in the buses, etc.
- Collected information can be used to infer likely causes behind the observed bus delays, the bottleneck traffic junctions, heavy traffic time slots, and make appropriate scheduling recommendations.
- Information about characteristics of the road traffic patterns, route delays, traffic growth trends can be of great help in proper planning. The collected information can also be used to perform *what-if* analysis to analyze the effect of various likely scenarios such as the effect of adding more buses, the effect of making changes in a road (widening the road, adding fly-overs, constructing dedicated bus-lanes) on traffic and delays, to name a few.

In this paper, we propose the use of advanced wireless technologies for automated monitoring of bus-operations. We argue that powerful analysis can be performed with simple and low cost infrastructure. For doing above analysis and for generating essential information, we need to track the bus on the route. There are multiple choices of wireless technologies that could be used for the purpose of tracking movement of buses such as GPS (Global Positioning System) with GSM [2], RFID (Radio Frequency Identifier) [3] and WSN (Wireless Sensor Network). Table I presents a qualitative comparison of the above three technologies. GPS+GSM, requires deploying GPS and GSM infrastructure on each bus. This is costly in comparison with the RFID and WSN infrastructure. RFID technology works best in short range communication scenarios and is very sensitive to interference resulting into an infeasible solution for road conditions. In [4] authors have enhanced reading range of RFID by incorporating WSN. [5] and [6] propose use of WSN with traffic signaling system to get maximum clear way for buses. We propose to use wireless sensor network as a feasible technology option for its low cost and adequate location estimation accuracy.

The main contribution of this paper is in demonstrating the use of wireless sensor networks to monitor bus operations. We argue that the minimal information

Technology	Accuracy	Deployment of Infrastructure	Interference	Cost
GPS+GSM	Good	Already available	Low	High
RFID	Good	Required	High	Low
WSN	Good	Required	Low	Low

TABLE I
COMPARISON OF VARIOUS POSSIBLE TECHNOLOGIES.

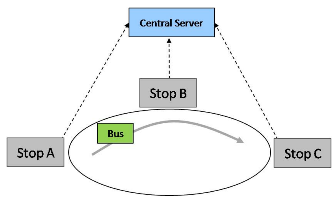

Fig. 1. Design of the proposed approach.

required to perform above analysis is the record of arrival time of the buses at bus-stops. Hence, the minimal setup required is to have a way for the buses and bus-stops to communicate and record the arrival time of the bus at the bus-stop. Currently in our proposed WSN, sensors are not sensing anything, however later we will provide more functionality like pollution monitoring, road health monitoring, weather monitoring etc, using appropriate sensors. Various design decisions need to be made while using WSNs for bus networks, such as how the buses and bus-stops should communicate, should the monitoring data be stored in a centralized or distributed manner, where to perform the analysis of the monitoring data, etc. We present various engineering design choices of this setup and argue for a suitable design.

The monitoring data thus collected provides information about arrival time of buses at bus-stops all over the deployment area. We next present algorithms that perform various analysis operations on the collected data. We present a proof-of-concept analysis by presenting two algorithms: (a) We present algorithms for detection of bus delays and localization of the likely cause of the delays. (b) We present algorithms to predict the likely arrival-time of buses at the bus-stops. For this purpose, we build models of traffic properties and their effect on latency.

II. WIRELESS SENSOR NETWORK DESIGN

Let us first take a simplistic example of a bus running in a loop visiting a fixed number of bus-stops. We present this setup in Figure 1.

- Sensors are placed on the bus and bus-stops. Bus-stops are directly connected to a central server e.g. through a wired network. Bus and bus-stops communicate through the wireless sensors.
- Bus sends beacons periodically. When the bus comes in the communication range of a bus-stop, bus-stop receives the beacon and records the time of receiving the beacon as the arrival time of the bus. Bus-stops periodically update the central server with the latest monitored information.
- The central server receives monitoring records from all bus-stops. All analysis is performed at the central server. Central server builds the model of traffic-patterns and delays on the basis of historical data and periodically updates this model based on received updates. This model is then used to perform various analysis operations such as localization of likely causes of delays, prediction of bus arrival time, etc.
- The bus-stops are periodically updated with relevant analysis results. Bus-stops and eventually buses then display this information such as current bus location, expected arrival time, congested roads, etc.
- Sensors on the buses and bus-stops do not need to be active all the time. Central server can also predict communication between bus and bus-stop using which effective sleep schedules can be built.

A. Design choices

We next present a systematic exploration of various design choices. The variations can be primarily in the following areas:

Analysis location: In the above setup the analysis was performed at the central server. An alternate choice is to perform analysis in a distributed manner at the bus-stops or buses.

Medium of communication to the server: In the above setup, bus-stops communicate with the central server to send monitoring updates and receive relevant analysis results. An alternate strategy is to use buses to exchange data between bus-stops and central server.

Beacon sender entity: In the above setup, bus sends beacons and the bus-stop listens for beacons. An alternate approach is to have bus-stops send beacons or have both buses and bus-stops send the beacons.

We use following notations: B for bus, S for bus-stop, SB for both bus and bus-stop, and CS for central server. We represent a design choice using a 3-tuple notation *Analysis location/Medium of communication to server/Beacon sender*. The above explained design will then be represented as *CS/S/B* - analysis is done at central server, the data is communicated through the bus-stop to the server, and the bus sends beacons that are heard by bus-stops.

We next present analysis of various design choices. We evaluate various design choices on the basis of following two parameters: (a) *Support for analytics applications:* We evaluate the designs from the point of view of how it can help to run various analytics applications on the buses or bus-stops. Some examples of the applications are: displaying on the bus-stop the expected time of arrival of next bus, displaying if the bus is delayed or not, displaying the congested link on the route causing the delay, etc. Similarly, buses can have displays showing the expected time of reaching a particular bus-stop, the delayed status of the bus and the other buses, the congested link on the route, etc. (b) *Ease and accuracy of sleep schedules:* We propose to make the sensors on the buses and the bus-stops sleep periodically to avoid them interfering into other wireless communication taking place in their vicinity. We consider how various designs would make communicating new sleep schedules efficient and accurate. We present two of the important models below.

CS/S/SB and S/S/SB: In this design, bus-stops are connected to CS. Both bus and bus-stop can send beacon. Location information of bus is generated on both (bus and bus-stop). Bus-stop would receive its analysis information from CS for its application and sleep schedule, whereas the bus would get its data from CS through bus-stop to run its application. Sleep schedule of the bus is maintained by the bus-stop i.e. bus-stop informs the bus its sleep duration till the next bus-stop. A drawback in this scheme is that packet collisions would occur when more than one bus is simultaneously trying to communicate with the bus-stop. This paper is based on the CS/S/SB design.

SB/SB/SB: Here both bus and bus-stop communicate to CS. Bus and bus-stop would perform analysis such as updating traffic model and prediction on their own. As they have sufficient information to run their application, they only need updates of bus location to properly run their application. So they would communicate very small amount of data (beacon and ACK) to each other whenever they come within range of each other. Due to the less data transfer between bus and bus-stop the duration of communication is small and there is less chance of collision of data packets.

III. Proposed approach for Data Analysis

We next demonstrate two analysis operations as examples of analysis that can be performed using the monitoring logs. Every bus and bus-stop is identified with an identifier denoted by BusID and BusStopID respectively. Bus stop communicates a log containing tuples of the form $< BusStopID, BusID, ArrivalTime >$ to the central server. From these logs, the analysis engine on the central server derives various information. For instance,

Round trip time of a bus: The time taken by the bus to complete one trip on a route can be computed. A time series of round trip time of each bus can be derived.

Link delay: Time spent by a bus on a link can be computed, where link refers to the path segment between two consecutive bus stops. A time series of link delay can be obtained for each link on the route.

A lot of analysis operations such as what-if analysis, impact analysis, capacity planning, fault localization, etc. can be performed using this information. Due to lack of space, in this paper we present two such analysis operations as examples, namely (a) fault detection and localization and (b) prediction. Below we first present details of the models that are built using these time-series. We then address the problem of detection and root-cause analysis of round-trip delays. Followed by that, we address the problem of prediction of the time of arrival of a bus.

A. Model building

We build a model for round trip time of a bus that captures the normal behaviour or the expected time required for a bus to complete one round trip in presence of traffic. Similarly we build a model for the link delay that captures the normal behaviour or the expected time required for any bus to travel on a link.

The round-trip delays and link delays keep changing across a day. For instance, early morning and late night latencies are very different from the latencies at peak rush hours such as late morning or early evening. We hence propose to identify steady states and build a temporally changing model that consists of multiple models for different steady state time intervals.

B. Steady state model

For a given measure (round trip delay or link delay), the model for steady state time interval needs to capture the normal behavior of the interval. We represent the model of a steady state using the mean and standard deviation of the steady state time-series. The model represents the expected normal values of the measure *m* (either round trip time or link delay) for a steady state interval *s* with the following equation:

$$M_m^S = [\mu - l * \sigma, \mu + l * \sigma], \quad (1)$$

where μ and σ represent the mean and standard deviation of measure m in steady state S. The variable l represents a threshold that represents the tolerance to noise and fluctuations. A value of $l = 1$ indicates that the values within the interval $[\mu - \sigma, \mu + \sigma]$ are considered normal. Higher the value of l, larger is the tolerance to noise and fluctuations to be considered as normal behavior.

C. Temporal model:

A temporal model for a measure m captures the properties of the measure m for an entire day. As the measure observes different behavior over the entire day, a temporal model consists of multiple steady-state models. Each steady state represents normal values of the measure m for a time-interval within the day. A temporal model for measure m can be represented with the following equation:

$$M_m^T = [M_m^{S_1}, M_m^{S_2}, \ldots, M_m^{S_k}], \quad (2)$$

where S_1 to S_k are k steady states in a day. One way to identify steady state intervals is to divide entire day duration into one-hour intervals, build steady-state models for each interval, and then merge consecutive intervals that have similar steady state models. Two steady state models are said to be similar if the mean value of one steady state model lies in the range of other steady state model.

We also plan to explore seasonal models that capture traffic across an year in different seasons. Further, capturing trends would help authorities to plan for long term development of infrastructure.

D. Detection and root-cause analysis of round trip delays

We next use the above defined models to detect delays in the round trip time of a bus. We then explain how the above models can be used to find the likely links that are causing the increase in the round trip time.

Detection of increased round trip time: Detection of the presence of abnormal delay in the round trip time of a bus b, on receiving the round trip time value m_{obs} for a time interval t, is done as follows:

1) Identify the temporal model M_m^T representing the round trip time measure m for bus b;
2) From the various steady state intervals in the temporal model M_m^T, identify the steady state interval S_i that contains the time interval t.
3) Identify the range of normal values, defined as window_size, of steady state model in the interval S_i, ($[\mu - l * \sigma, \mu + l * \sigma]$).
4) Detect a delay in round trip time if m_{obs} is not in the expected range of normal values.

Localization of most likely delay causing links: On detecting the presence of a significant delay in the round trip time m of a bus b in time interval t, the localization of the links that are most likely causing the delay can be done as follows:

1) Identify the set of links L_b that consist the route of the bus b.
2) Analyze each link $l \in L_b$ to have observed significant delays in time interval t. This can be done using the link latency value observed in the time interval t and steady state link delay model built for the link l.
3) Declare the links that observe most significant increase as the likely causes.

In the list of most likely delay causing links, it might be possible that the actual congestion occurs in one link and the traffic on this link overflows into adjacent links causing congestion in them. We can find links where actual congestion occur from the set of congested links, by looking on the correlation of time series of links.Due to lack of space we are not giving the details of algorithm here.

E. Prediction of the time of next arrival of a bus

We next use the path and link models to predict the time of next arrival of a bus b at a bus stop s. Using the temporal models for paths and links, the arrival time of a bus at a bus stop can be predicted in a straight-forward manner as follows:

1) Identify the set of links L_b that consist the route of the bus b.
2) From the temporal models of the links in L_b, identify the steady state model that best contains the current time.
3) For each link $l \in L_b$, from the steady state models identify the expected value range of link delay.
4) Add the link delays of all links in L_b to the time when the bus b was last observed at bus stop s to compute the expected next arrival time of the bus b at stop s.

IV. Experimental Evaluation and Results

We perform simulations using CSIM [7]. The algorithms are implemented in R [8]. The topology used is shown in Figure 2(a). Simulation consists of eight bus stops namely A, B, C, D, E, F, G and H and three buses, each running on one route namely $R1 : ABDE$, $R2 : AGFE$ and $R3 : AGHCDE$. Each link between the two consecutive bus-stops has a preset link delay that is the time a bus takes to travel that link in absence of traffic congestion. Random variations in the link delays are added to carry on experimentation. We simulate bus stops as nodes and buses plying on routes as data packets in CSIM and collect arrival time of bus on a bus stop.

The data from all bus-stops is collated at central server. We build model for link delay for each link and model of round trip time of each bus from the collated data. Once the models are built, we use these for detection of

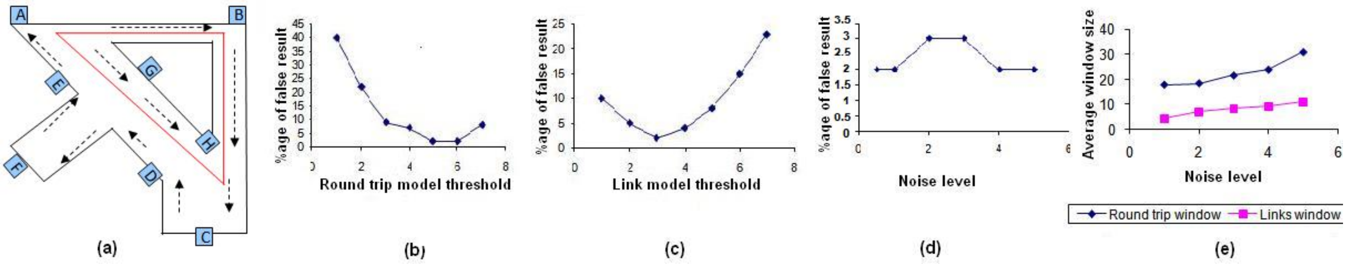

Fig. 2. (a) Topology used for the experiments. (b)Effect of round trip looseness on Delay Detection result, (c)Effect of link looseness on Localization of cause result, (d)Effect of different level of noise in data on Delay Detection results and (e) On Window size of model.

delayed buses and for identifying causes of the delay. We use proposed delay detection algorithm on the round trip time of a bus. In case of detection of delay in the bus arrival time, the algorithm identifies congested link with larger than normal delay as the cause. We experimented with multiple instances of delayed buses and identified cause of the delay.

We carried out experiments with different values of threshold l. As explained earlier, the threshold value represents the tolerance to noise and fluctuations. If value of l is kept very small there can be instance when a bus is identified as delayed when it is not actually delayed. We call such instance a false-positive. Similarly, we define false-negative as an instance when a delayed bus is not detected as delayed by a large value of l. Initial experiments show that a small value of l for link model and a large value of l for round trip model would result in fewer false positives and false negatives. This can be seen in Figure 2(b). The figure shows that on increasing the value of l of round trip model, percentage of false result (a sum of percentage of false positive and negative both) decreases due to decrease in number of false-positives but after certain value of $l(= 5)$ the percentage of false results increases due to increase in number of false-negatives. Similar trend can be seen for link model in Figure 2(c).

A. Sensitivity Analysis

In this section we evaluate how sensitive our algorithm is to various levels of noise in the data. Following are three parameters used for sensitivity analysis with their default values:(1) Round trip threshold $(rl) = 5$, (2) Link model threshold $(ll) = 3$.

Figure 2(d) shows that increase of noise level in data doesn't bring significant change in the percentage of false results. The reason is that on increasing noise level, threshold $l * \sigma$, increases which in turn also increases window_size to $2 * l * \sigma$ (Figure 2 (e)). Increase in noise level results in large variation in observed values (m_{obs}). However, the increase in window size increases the tolerance to these noise variations there by preventing the false positives.

V. Conclusion

In this paper, we explore use of wireless sensor networks (WSN) for improving bus transportation system. We demonstrate the use of WSN for monitoring bus operations. We propose to use minimal information viz. record of arrival time of the buses at the bus-stops for performing analysis. We present various design choices available using wireless sensor networks. We then propose models to represent round trip times and link delays. In order to demonstrate how the collected logs can be used for analysis, we present two example analsis operations viz. detection and root cause analysis of bus delays, and predicted of the time of bus arrival.

References

[1] A. Jogalelar, "Rationalising and Reforming Bus routes - case study," in *http://praja.in/en/discuss/forums/2008/07/rationalising-and-reforming-bus-routes-case-study*, 2008.

[2] J. Owusu, F. Afukaar, and B. E. K. Prah, "Towards Improving Road Traffic Data Collection: The Use of GPS/GIS," in *FIG Regional Conference*, 2006.

[3] A. Rahmati, L. Zhong, M. Hiltunen, and R. Jana, "Reliability Techniques for RFID-Based Object Tracking Applications," in *DSN*, 2007.

[4] B. A. Hatem and H. Habib, ""Bus Management System Using RFID In WSN"," in *EMCIS*, 2010.

[5] W. MA and X. YANG, ""Design and Evaluation of an Adaptive Bus Signal Priority System Base on Wireless Sensor Network"," in *IEEE ITSC*, 2008.

[6] N. Hounsell, B. Shrestha, F. McLeod, S. Palmer, T. Bowen, and J. Head, ""Using global positioning system for bus priority in London: traffic signals close to bus stops"," in *IEEE IET*, 2007.

[7] H. Schwetman, "CSIM: A C-Based, Process-oriented Simulation Language," in *Winter Simulation Conference*, 1986.

[8] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.