IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# AUTOMATIC TOLLGATE BILLING SYSTEM USING RSSI

Jayasree T[*], Esther Preethi S[**], Abisha A[**], Lingaprema S[**], Gibbson Clive Magesh E[**]

[*] *Assistant Professor, ECE Department, Govt. College of Engineering, Tirunelveli*

[**] *UG Students, ECE Department, Govt. College of Engineering, Tirunelveli*

***Abstract*— Electronic toll collection (ETC) is a technology, enabling the electronic collection of toll payments. With the development of the freeway, at present Electronic Toll Collection (ETC) system more and more is used in the freeway management system, which can finish identification of the vehicle and vehicle toll without parking automatically. With the development of the short wireless communication technologies, such as RFID, WiFi, UWB and Bluetooth reveal the great application prospects in the freeway ETC system and promote the research and development of new ETC system. Radio Frequency Identification (RFID) is a grown automatic identification technology and nowadays widely used in the ETC system with the advantage of the short transmission distance, no-touching and fast response. This paper put forward the RSSI technology to use in the vehicle localization to improve the efficiency of the freeway ETC system. RSSI (IEEE 802.15.4 standard) is a rising wireless network technology which is of short space, low complicity, low power consumption, low data rate and low cost.**

## I. INTRODUCTION

In this paper we have anticipated an enhanced version of tollgate billing system. a proficient way of communication link between wireless channels to simplify vehicle monitoring. Once the motorist comes into the chargeable zone, the ZigBee transceiver receives the acknowledgement that the vehicle has arrived the chargeable region specific lane then eventually the unique Id of the motorist has been recited via on board unit. Researchers have implemented a variety of techniques to accomplish this, such as the inclusion of ultrasonic sensors for measure the time of flight between transmitting and receiving sound, directional antennas for broadcasting exclusively in a certain direction and laser scanners which project a beam of light across a distributed network with light sensors equipped to each node, it is more efficient if the nodes can determine their own locations since networks tend to consist of hundreds of devices, and it would be incredibly tedious to prepare each one with an individual location. Location estimation is a common application of wireless sensor networks (WSN).

In WSNs, it is desirable to keep each node as inexpensive and small as possible and increase battery life as long as possible. The inclusion of additional hardware for localization purposes makes these constraints more difficult to comply. It would be preferable to use radio propagation to estimate distance as the network is already using wireless transmissions to communicate. There are two methodologies primarily used to interpret distance from wireless transmissions: time delay on arrival and signal strength attenuation. There is one major factor that complicates distance estimation from wireless transmissions - the multipath effect. The multipath effect occurs when a wireless signal finds multiple paths from the transmitter to the receiver by reflecting off of other objects within the environment, such as walls and floors. This causes the signal to attenuate or amplify depending on the phase of the reflected signals and the line-of-sight signal when combined at the receiver. The multipath effect is more prominent in indoor environments, however, it still exists in outdoor environments as well, but usually the signal scattering is more predictable. Time delay on arrival techniques are less sensitive to this effect as they do not interpret distance from attenuation as signal strength-based techniques do. Time delay techniques also require precise oscillators, as well as clock synchronization across the network, which demands higher quality and more expensive hardware. Signal strength attenuation is represented by received signal strength indicator (RSSI) in negative decibel mill watts (-dBm). In free space, signal strength attenuates logarithmically with respect to the distance between transceivers.

The RSSI has often been used in location estimation and tracking applications. Signal strength between wireless transceivers degrades according to how much distance is between them, allowing the distance to be estimated. With this estimated distance between wireless devices, location estimation techniques can be used. However, one of the largest hindrances in using RSSI for distance estimation for indoor applications is reflections from multipath effects from the wireless signal.

The Kalman filter is a recursive Bayesian filter, which models the noise of each input as a Gaussian distribution. Using the estimated motion and distance within a distributed, wireless network, the effects of multipath fading can be reduced for distance estimation with Kalman filtering.

Source localization using compressed sensing in the context of indoor RF-based AoA localization has been theoretically studied in the past. These theoretical analysis has revealed that using compressive sensing techniques for small RF antenna arrays is feasible.

## II PROPOSED METHODOLOGY

In the proposed model, the vehicle is enabled with an on-board unit, which contains the vehicle ID and information. Since WSN can address 16-bit addressing, more number of vehicles can pass freely. Uses bloom filter for network traffic problem. Here once the vehicle reaches the toll gate, Received Signal Strength Indicator (RSSI) shows the RF signal strength in terms of dBm. If the strength value is -50 to -59dBm, it alters the vehicle owner that the toll gate is about to reach. It also gives a voice message as “Welcome to Toll gate”. Now the user has to select the trip as, one way or two ways through the keyboard interface. User toll gate amount will be displayed according to the trip selection. As the vehicle reaches the toll gate, the RSSI value increases to -30 to -39dBm.

Now it alters the vehicle owner that it has reached the toll gate. Radio frequency Identification tag will automatically detects the vehicle and connects to the server by Zigbee wireless communication network. After transaction is made, the barrier automatically opens and the vehicle crosses the toll gate without waiting in the queue lane. Once the vehicle crosses the toll gate, voice message indicates that the vehicle has crossed the toll gate. User can cross the toll gate for maximum of two times with zero balance in the user account.

In this approach, once the end user (motorist) comes to chargeable zone, the Zigbee transceiver receives the signal that the vehicle has entered the chargeable zone particular LANE, then automatically the unique id of the motorist (Vehicle ID) has been read via on-board unit, which has been attached to every vehicle and collects the daily road pricing charge. Here, two nodes are used. Each Node contains two individual tags to identify the authentic person. Tags are based on passive RFID (Radio-frequency identification), so that one tag of one node cannot be used in another node. Each Tag has unique code number for authentication. If we use another tag instead of specified tag, then buzzer will produce a sound and usage of tags can be monitored in host.

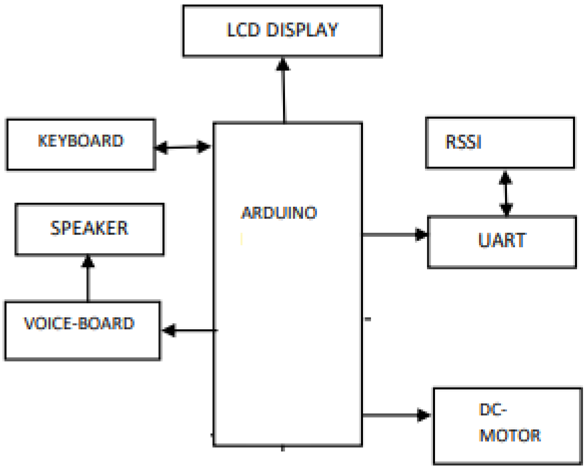


Figure 1 Proposed Model

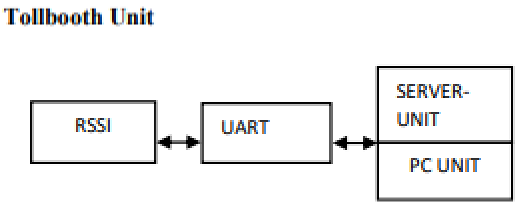


Figure 2 Tollbooth unit

**Arduino UNO:**

This is the latest revision of the basic Arduino USB board. It connects to the computer with a standard USB cable and contains everything else you need to program and use the board. It can be extended with a variety of shields: custom daughter-boards with specific features. It is similar to the Duemilanove, but has a different USB-to-serial chip the ATMega8U2, and newly designed labeling to make inputs and outputs easier to identify.

**Arduino Mega 2560:**

*A larger, more powerful Arduino board. Has extra digital pins, PWM pins, analog inputs, serial ports, etc.* The version of the Mega released with the Uno, this version features the Atmega2560, which has twice the memory, and uses the ATMega 8U2 for USB-to-serial communication.

## III SOFTWARE DESCRPTION

The Arduino Software (IDE) makes it easy to write code and upload it to the board offline. We recommend it for users with poor or no internet connection. This software can be used with any Arduino board.

There are currently two versions of the Arduino IDE, one is the IDE 1.x.x and the other is IDE 2.x. The IDE 2.x is new major release that is faster and even more powerful to the IDE 1.x.x. In addition to a more modern editor and a more responsive interface it includes advanced features to help users with their coding and debugging.

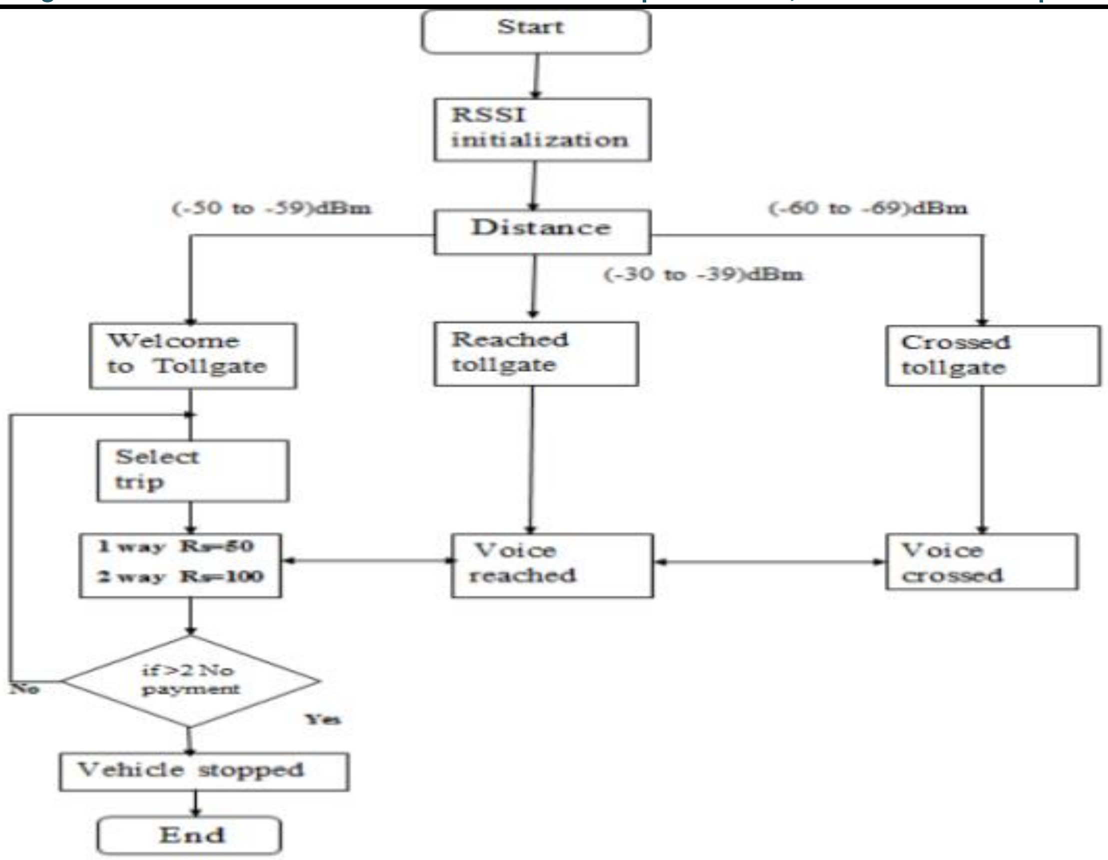


Figure 3 Flowchart

The following steps can guide you with using the offline IDE (you can choose either IDE 1.x.x or IDE 2.x):

**1.** Download and install the Arduino Software IDE:

- Arduino IDE 1.x.x (Windows, Mac OS, Linux, Portable IDE for Windows and Linux, ChromeOS).
- Arduino IDE 2.x

**2.** Connect your Arduino board to your device.

**3.** Open the Arduino Software (IDE).

**The Arduino Integrated Development Environment** - or Arduino Software (IDE) - connects to the Arduino boards to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called **sketches**.

*Using the offline IDE 1.x.x*

The editor contains the four main areas:

**1.** A **Toolbar with buttons** for common functions and a series of menus. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

**2.** The **message area**, gives feedback while saving and exporting and also displays errors.

**3.** The **text editor** for writing your code.

**4.** The **text console** displays text output by the Arduino Software (IDE), including complete error messages and other information.

The bottom right-hand corner of the window displays the configured board and serial port.


The Figure 4 Arduino Software IDE

Now that you are all set up, **let's try to make your board blink!**

5. **Connect your Arduino** or Genuino board to your computer.

**6.** Now, you need to **select the right core & board**. This is done by navigating to **Tools** > **Board** > **Arduino AVR Boards** > **Board**. Make sure you select the board that you are using. If you cannot find your board, you can add it from **Tools** > **Board** > **Boards Manager**.

**7.** Now, let's make sure that your board is found by the computer, by **selecting the port**. This is simply done by navigating to **Tools** > **Port**, where you select your board from the list

**8.** Let's **try an example**: navigate to **File** > **Examples** > **01.Basics** > **Blink.**

**9.** To **upload it to your board**, simply click on the arrow in the top left corner. This process takes a few seconds, and it is important to not disconnect the board during this process. If the upload is successful, the message "Done uploading" will appear in the bottom output area.

**10.** Once the upload is complete, you should then see on your board the yellow LED with an L next to it start blinking. You can **adjust the speed of blinking** by changing the delay number in the parenthesis to 100, and upload the Blink sketch again. Now the LED should blink much faster.

## IV CONCLUSION

In this paper, the major objective has been to quantify how good and accurate is the RSSI model in a wireless sensor network to estimate the location of a cooperative target. The system proposed here doesn't need smart card. The vehicle is enabled with an on-board unit that contains the vehicle id and information, where tollgate payment is carried out at particular RSSI value. Results are encouraging and are able to achieve an accuracy of nearly 1.1 meters with 90% probability in indoor environment. If have a good quantification of the signal strength model, it can be implemented for location estimation scheme on this basis.

## REFERENCE

[1] Aaron Ault, Edward Coyle, and Xuan Zhong. K-nearest-neighbor analysis of received signal strength distance estimation across environments. Proceedings of the First Workshop on Wireless Network Measurements, April 2005.

[2] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. INFOCOM (2), pages 775– 784, 2000.

[3] E. Elnahrawy, Xiaoyan Li, and R.P. Martin. The limits of localization using signal strength: A comparitive study. IEEE Conference on Sensor and Ad hoc Communications (SECON), 2004.

[4] Marko Helen, Juha Latvala, Hannu Ikonen, and Jarkko Niittylahti. Using calibration in rssi-based location tracking system. Proceedings of the 5 th World Multiconference on Circuits, Systems, Communications and Computers, July 2001.

[5] J. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Towards sophisticated sensing with queries. Proceedings of 2nd International Workshop on Information Processing In Sensor Networks (IPSN), 2003