International Journal of Advance Research In Science And Engineering http://www.ijarse.com
IJARSE, Vol. No.4, Special Issue (01), March 2015 ISSN-2319-8354(E)

DESIGN AND IMPLEMENTATION OF ANTI-COLLISION RAILWAY SYSTEM USING HYBRID OF ZIGBEE NETWORK AND INTERNET OF THINGS

Sahil Dange

E&C Department, Pune University, (India)

ABSTRACT

To avoid accidents of railways and save life's of many people it is very important to have a robust, automatic and scalable solution which would improve signaling system and reduce the chance of accidents due to human errorsand old railway signaling methods. This paper discusses the architecture and implementation of Anti-Collision railway system using ZigBee and RFID network and Azure cloud service as a backend system. The proposed system leverages the scalability of ZigBee network and its self-healing property with Azure stream service as its backend. Each rail track would be uniquely identified by maintaining a RFID tag to it. The train will have a small embedded device which would communicate with Azure cloud service, read the RFID tag of the track on which train is running and make important decisions based on the information it gathers from cloud service. The train network could consist of ZigBee coordinator residing at railway control station, ZigBee router residing on train and ZigBee end device residing at signal posts. All railway control stations would be connected to each other via Azure cloud service thus forming dense centralized intelligent railway system.

Keywords: Asynchronous Message Queuing Protocol (AMQP), Azure Event Hub, Internet of Things (IOT), Personal Area Network (PAN), Radio Frequency Identification (RFID), Zigbee Coordinator

I. INTRODUCTION

The train accident generally happens because of human errors and machine failures. The Indian railway losses huge amount of money due to cancellation of trains during winter every year. Also it is quite difficult to run trains in the winter every year. The main reason behind this problem is fog. Indian railways has worked on many technologies to overcome the hindrances. The ACD (Anti-Collision device) [1, 2] is developed by konkan railways which uses the GPS technology for tracking the position updates of the train. The drawback of ACD is that it could not detect the rail tracks separated by a distance of 10-15 feet because of limitations of accuracy of GPS in our country. The ACD does not take into account factors based on environment. As a result accident due to other factors like collapsing of bridge or derailment cannot be overcome. All these above limitations are taken into consideration and a new architecture of the system is proposed in this paper and its prototype is implemented.ZigBee technology [3,4] was selected as a last mile communication device because of its highly scalable, self-healing, low power and its unique radio properties. Thus ZigBee is a perfect candidate suitable in wireless communication for embedded system design point of view. Also ZigBee supports large number of nodes, easy to deploy, have a very long battery life, secure, low cost and can be globally used. To support, monitor and control such a large infrastructure which consists of train inventories, monitoring weather conditions, do intelligent rerouting of rails based on traffic and environmental conditions we will leverage Internet of Things technology which would connect to Azure cloud service. All the ZigBee devices connected on trains, bridges, signal post etc. will connect to Azure event hub via ZigBee gateway located at railway control station.

II. LITERATURE SURVEY

The existing conventional signaling system most of the times relay on the oral communication through telephonic and telegraphic conversations as input for the decision making in track allocation for trains. There is large scope for miscommunication of the information or communication gap due to the higher human interference in the system. This miscommunication may lead to wrong allocation of the track for trains, which ultimately leads to the train collision. The statistics in the developing countries showing that 80% of worst collisions occurred so far is due to either human error or incorrect decision making through miscommunication in signaling and its implementation. IR sensors are also used to identify the cracks in the railway. IR sensors have limitations due to the geographic nature of the tracks. The Anti-collision device system also is found to be ineffective as it is not considering any active inputs from existing Railway signaling system, and also lacks two ways communication capability between the trains and the control centers or stations. Later geographical sensors have also been used which makes use of satellites for communication. But the system is costly and complicated to implement. At present laser proximity detector is used for collision avoidance, IR sensors identifies the cracks in the railway track and gate control is done by manual switch controlled gate. But there is no combined solution for collision between trains, train derailment in curves and bends and the automatic control of railway gate.

2.2 History of Anti-Collision Device for Train

The ACD Network is a Train Collision prevention system invented by RajaramBojji and patented by Konkan Railway Corporation Limited (a public-sector undertaking of the Ministry of Railways, Government of India). ACDs have knowledge embedded intelligence. They take inputs from GPS satellite system for position updates and network among themselves for exchanging information using their data radio modems to take decisions for timely auto-application of brakes to prevent dangerous 'collisions', thus forming a 'RakshaKavach' (meaning a '(Train) Safety shield').ACDs fitted (both in locomotive and guard's van of a train) act as a watchdog in the dark as they constantly remain in lookout for other train bound ACDs, within the braking distance required for their relative speeds. They communicate through their radios and identify each other. If they happen to find themselves on the same track and coming closer to each other, they automatically restrain and stop each other, thereby preventing dangerous head-on and rear-end collisions. Loco ACD of a train also applies brakes to reduce the train speed either to 15 km/h if on approach it receives a message from other train bound ACD that has stopped in a block section on adjacent track (and driver of that train has yet not communicated that things are 'Normal') or to bring the train to a stop if train bound ACDs of other train are radiating 'train parted' message thereby preventing dangerous side collision that may occur due to infringement of adjacent track by a stopped or a 'parted' train, respectively. ACD trials have recently been concluded successfully in Southern Railway. Further implementation on Indian Railway is awaited.

Level crossing: Loco ACD on receipt of 'Gate Open' input from Gate ACD (provided at non-interlocked level crossing), applies brake to regulate its train speed to 30 kmph or as per requirement previously set. Gate ACDs fitted at manned and un-manned level crossings also give audio-visual 'Train Approach' warning to road users if an ACD fitted train approaches them. Also if a Loco ACD receives manual 'SOS' message from other Train bound ACD or a Station ACD that is within three kilometers of its radial range; it applies brakes automatically to bring the train to a stop. The application of this Anti-Collision Device has been refined to not only prevent 'midsection' collisions but also to prevent their occurrences in 'station yards'. The newly engineered solution is integrated with the signaling systems and interlocking to react appropriately in case collision like conditions are perceived at the time of reception and dispatch of trains from a station, for e.g. while approaching a Station, Loco ACD also gives 'Station Approach' alert to the driver and regulates its train speed when it receives information from Station ACD to this effect, namely, either main-line is occupied by a train OR a load stabled on it and not fitted with ACD OR if route for train reception is not set for main line. Loco shed ACDs, Track-ID Assigning ACDs and Repeater ACDs further to strengthen the working of ACD Network.

ACD deficiencies: The ACD system is based on GPS based positioning and track detection. This had its inherent problems as with the GPS- Standard Positioning GPS service or Coarse acquisition (Precision positioning is only available in US for military use) the best possible horizontal accuracy is 10 m. This is inadequate for detection of rail tracks separated by a distance of 10–15 feet. ACD does not even have DGPS (Differential GPS that gives an accuracy close to 2.5m) and hence had errors in track detection using their patented Deviation Count Theory that worked in block sections but failed in station sections. The result was erratic braking that disrupted train movements and proved to be ineffective. Another design different to ACD was patented in 2001, named "Railway Collision Avoidance System" by an Indian inventor, IndranilMajumdar from Calcutta who was awarded the Texas Instruments Analog Design Challenge 2001 for this design and the patent granted in 2007.The design ruled out GPS as it was a 3rd party system (US based) and suggested track based sensors similar to RFID (Radio Frequency Identification Device) or Balise (similar to the EuroBalise). The design didn't receive much attention as the inventor had no railway background. However, the design concepts were eventually reflected in the TCAS design first released in 2008.After 7–8 years of problems with the ACD system, RDSO, Lucknow drafted the Train Collision Avoidance System (TCAS) specs. With amendments that followed and finally in 2012, the Ver3.1.1 spec. has been released (after joint consultation with companies manufacturing signaling equipment for the Indian Railways). The ACD system though in use with the Indian Railways, has its inherent problems in Station Sections due to its design concept of using GPS for track detection that is not viable.

The High Level Safety Review Committee at Mumbai on 12/13 January 2012 at the Western Railway HQ were sceptic of the ACD effectivity and were unanimous of having TCAS developed, as an open architecture system that shall offer multi-vendor operability without attracting any royalty unlike the ACD which is proprietary. TCAS being developed by qualified companies (RDSO approved and manufacturing railway safety systems that includes, Kernex Microsystems, Medha Servo Drives Hyderabad, Invensys Bangalore, Siemens, HBL Power Systems Ltd Hyderabad and others) in India selected by RDSO through an Expression of Interest (EOI), shall be an Indian Train Protection System offering collision avoidance and also offer many functionalities of the European Train Control System that includes prevention of Signal Passing at Danger (SPAD), Movement Authority and Control, Critical Train Data Recorder, advance in cab display of signals, advance alerts and warnings from Station sections, uploading of running train data to a Central Train Management System over

GSM-GPRS, etc.TCAS has been selected for implementation and it is going to cost very heavy to Govt as per this news report there would be a cost of 10 L INR per kilometer

III.PROPOSED SYSTEM

The architecture is divided into two parts.

1. Front End Architecture
2. Backend Architecture

3.1. Front End Architecture

Following components are the part of Front End Architecture:-

1. ZigBee networks
2. RFID tags
3. Environment sensors enable for IoT technology[8]

3.1.1 Zig Bee networks

Each railway control station will have one ZigBee coordinator which will form ZigBee networks with unique PAN ID.

Each train will have ZigBee routers which would join the ZigBee networks. ZigBee networks are formed with mesh topology. In mesh topology network coordinator is responsible for the network initialization and maintenance. Routers are used to extend the network. Thus a mesh network is called a self-healing network where the different devices performs the routing in the network and if one node fails another node can be used for delivery. Thus as shown each railway station may have their own ZigBee network. Each railway station will connect to Azure cloud service for intelligence railway management, real time event monitoring, machine learning and generating insights and reports. Each train will communicate to Azure via ZigBee router i.e. railway control station.

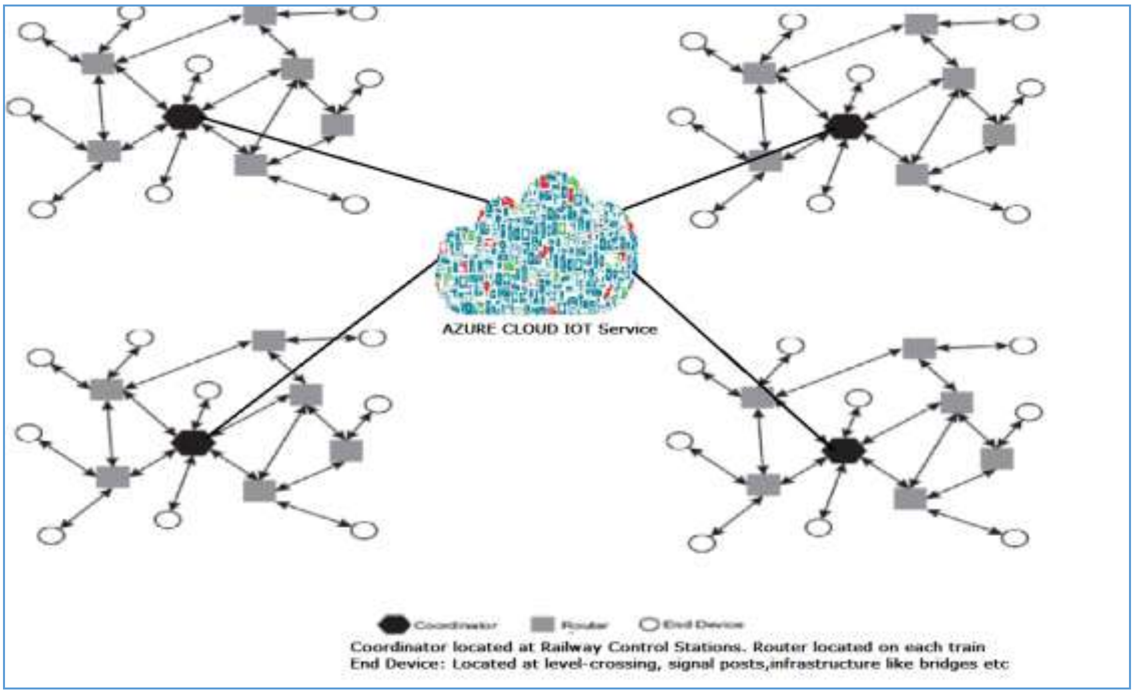

Fig 1: High Level Architecture of Proposed System

3.1.2 RFID Tags

The train detects the correct train track using radio frequency identity technology. Each rail track has a unique RFID tag attached to it. Fig 2 shows the fitment of RFID tags, ZigBee router, and ZigBee coordinator in the proposed system

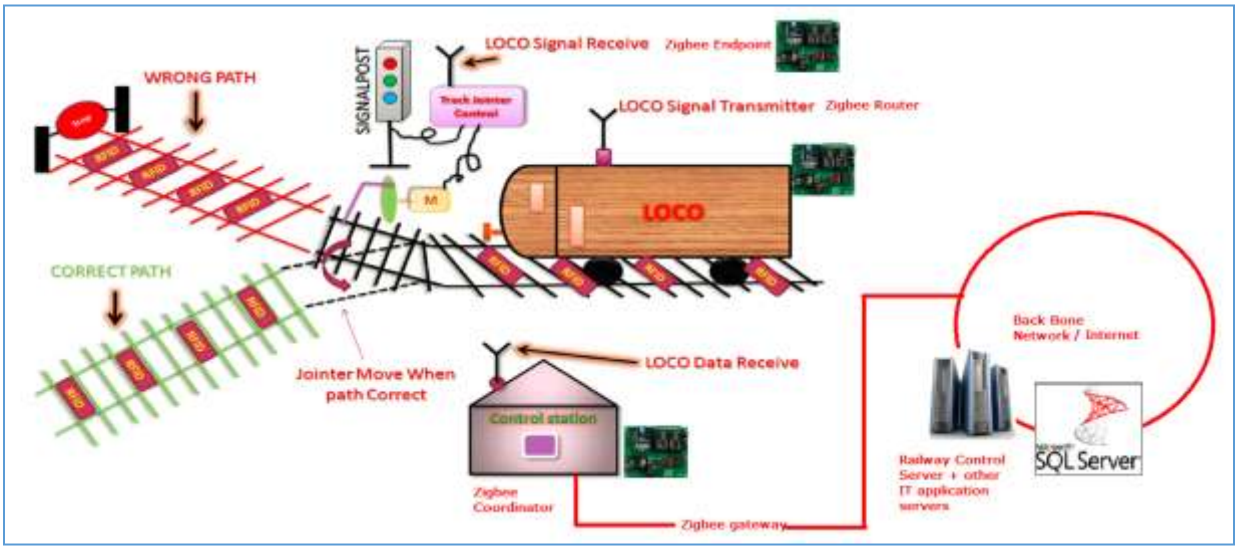

Fig 2: High Level Architecture of Proposed System

In this architecture control stations lookup the Azure backend database to get the correct track ID and transmits it to the all trains within its network. If the train is not on the correct track ID on which the train is supposed to run, the embedded system in the train would invoke automatic braking and provide audio-video alerts. The cloud service would not allow to run two trains on the same track in opposite direction within the same time frame as all intelligence of the system is built centrally at cloud service. The decisions are taken at the cloud side and are propagated down to the train via ZigBee networks. The train first discovers the ZigBee network and registers itself to ZigBee coordinator via nearby ZigBee devices located on another train or signal post. After network discovery, the train would read the track ID via RFID. The train would get all necessary information from cloud services and cloud service can respond on real time bases to the response it receives from all the control stations simultaneously.

3.1.3 Environment Sensors Enable for IOT Technology Connecting To Azure

All the end devices like temperature sensors, humidity sensors, wind sensors, fire-sensor,and ZigBee coordinator can communicate to the Azure via a gateway. The gateway converts the propriety protocol to IP protocol. So every device is assigned an IPv6 public IP. Thus device can communicate directly with each other and also with Azure calling this as a machine to machine networking terminology [9].

3.2 Backend Architecture

The ZigBee coordinator located at railway stations, and other devices like sensors connect to Azure event hub. Event hub of Azure comes under Azure stream line analytics. Heart of the proposed system is Azure NRT where the intelligence is stored. All the real time events are collected from the devices and stored in Azure SQL [5]. Transforms of the data are done using simple SQL queries and the necessary events are transmitted back to the end devices via Azure Event hub. Fig 3 shows the high level architecture of backend

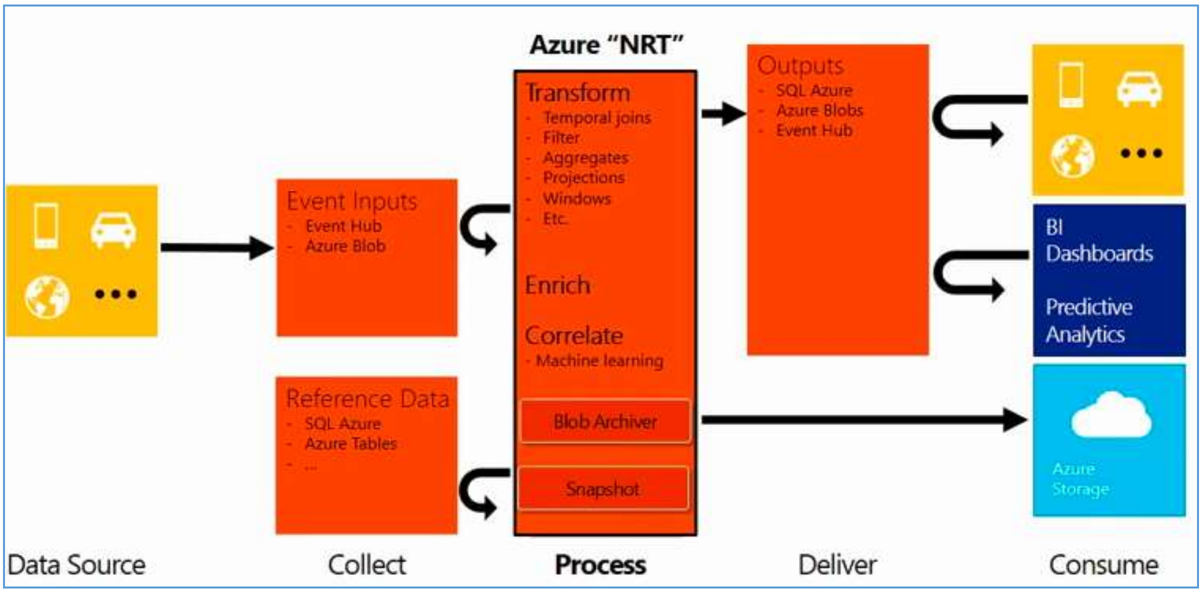

Fig. 3: High level architecture of backend system in Azure

ZigBee coordinator can connect to Azure via field gateways like Arduino Ethernet shield. Sensors can connect to the Azure event hub via various protocols like HTTP and Asynchronous Messaging Queue protocol. HTTP protocol is very insufficient as a messaging protocol, hence Microsoft's AMQP protocol [6] is used to transfer the data from end devices to cloud.

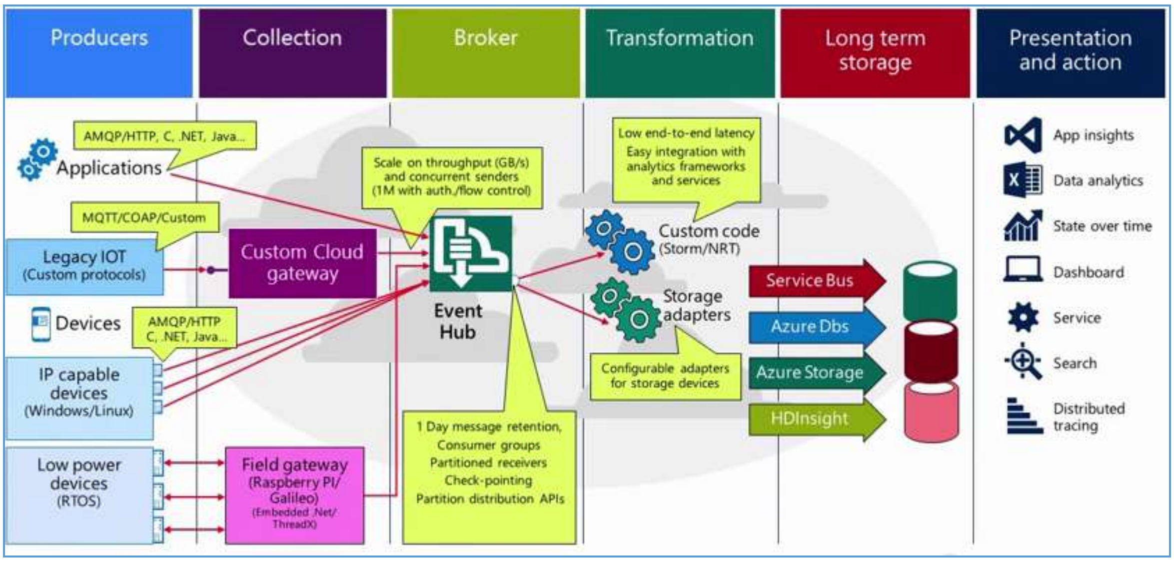

Fig. 4: Detail architecture of Azure event hub

IV. IMPLEMENTATION OF FRONTEND

We have implemented train module which consists of ARM controller LPC2148, LCD to display events and alerts/warnings, DC motor acting as train engine, power supply unit and a ZigBee router. The main work done by the microcontroller is to discover nearby ZigBee network and register to it, continuously monitor strength of the received signal by reading the RSS (Received Signal Strength) value and register to the ZigBee network having maximum RSS.

Microcontroller also processes the information received from Azure and the trains within its network. Fig 5 shows the high level block diagram of the implemented module for the train.

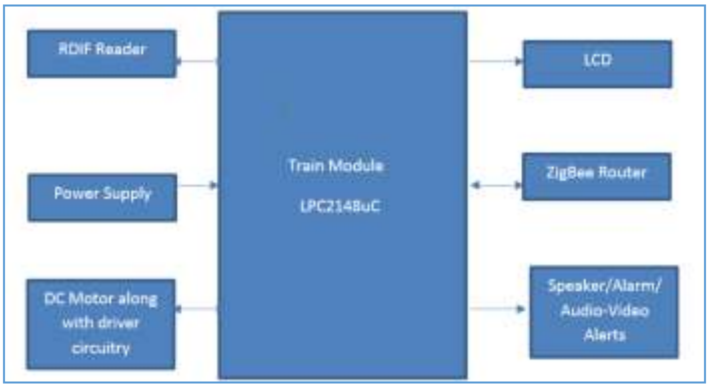

Fig. 5: Block diagram of module attached to trains

The module present at railway control stations have Arduino microcontroller and a ZigBee router connected to Arduino Ethernet shield acting as a gateway to Azure. Main purpose of this module is to initiate and form ZigBee network with unique PAN and is to interface with Azure cloud.

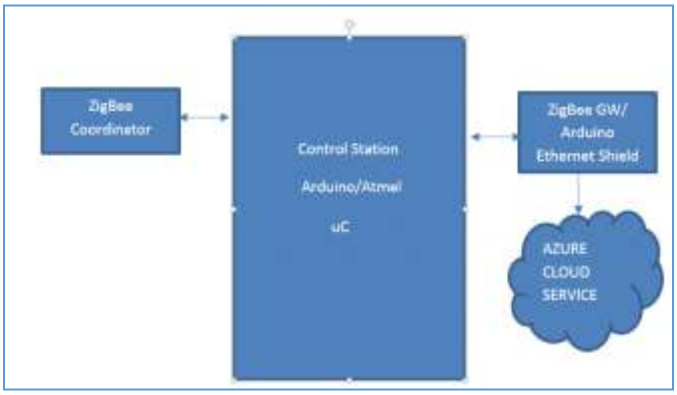

Fig. 6: Block diagram of module present at railway control stations

4.1 Workflow of the Train Module

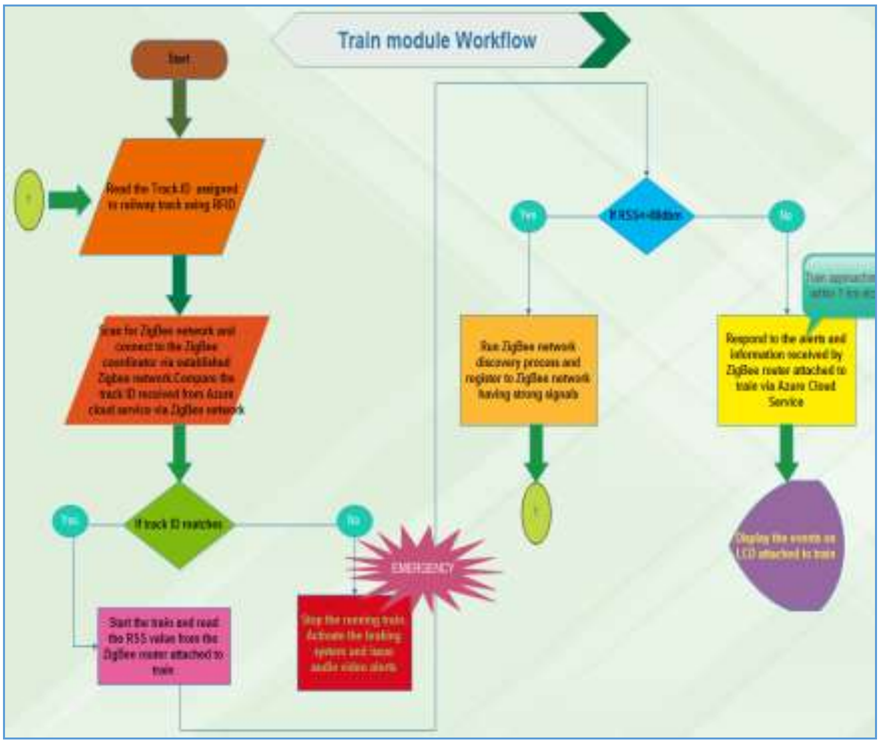

Fig. 7: Workflow for implemented train module

Workflow for train:-

1. Each train has a ZigBee router connected to it as shown in Fig 6 block diagram. The router will connect to ZigBee coordinator located at railway control station by scanning though the available ZigBee network.
2. Each railway control station is connected to Azure service bus using internet/MPLS or E-route provided by AT&T network.
3. ZigBee network would be set by ZigBee coordinator with some personal ID number. Trains falling under the given railway control station would join that particular ZigBee network.
4. When train modules star it would first read the underlying RFID tags attached to the railway track. Each railway track has unique identify identified by its RDIF tag.
5. Train will then query its current configuration i.e. destination it has to reach, path needed to be taken by it to reach destination etc. from railway control station. Railway control station will turn query the necessary train configuration from Azure cloud services.
6. The train module will then compare these two values received from control station and RFID. If the values are equal the train would start and check the quality of signal received for ZigBee network. If the signal value falls below -80dbm ZigBee router would rescan the ZigBee network and connect to the ZigBee network having stronger receive signal strength. Stronger signal strength is preferred to reduce bit errors and retransmission.
7. If the ZigBee signal strength is good then the train would listen to the events received to it and respond to it. Example if there collision like situation, the train would issue audio-video alerts, will invoke automatic braking system etc.

4.2 Workflow of the Signal Post and Level Crossing Module

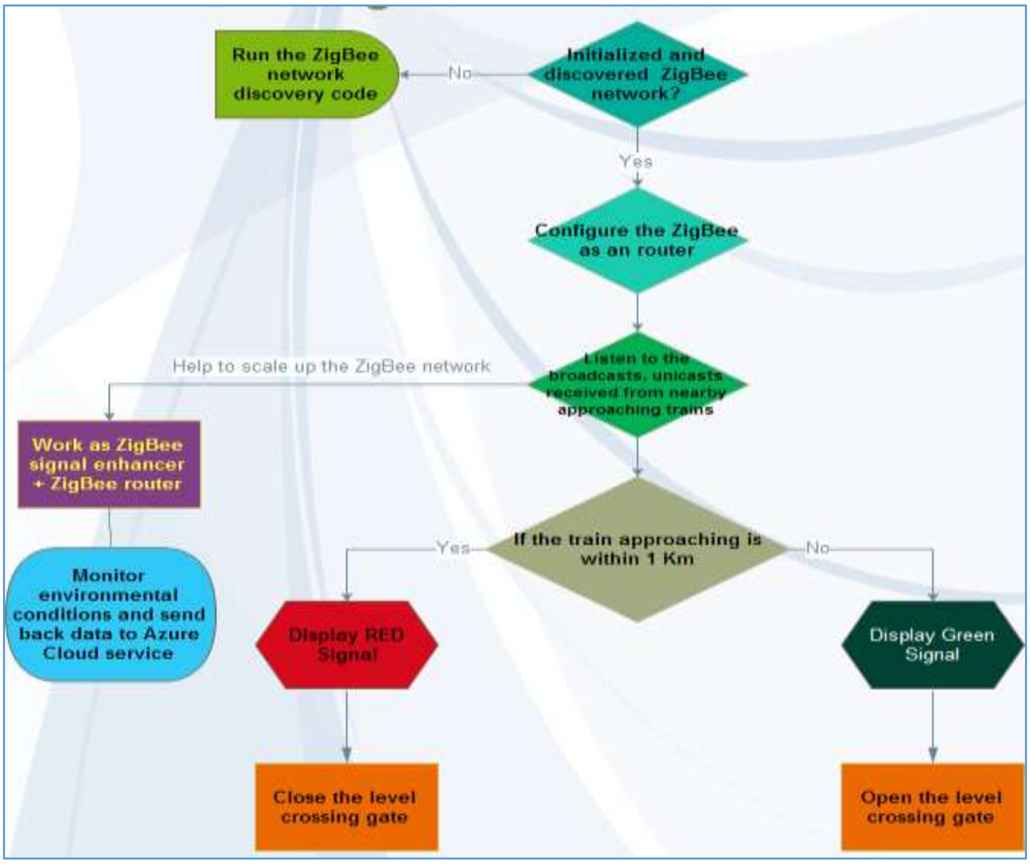

Fig. 8: Workflow for simulated train module

Workflow for Signal Post and level crossing module

1. Each signal post will be connected by a ZigBee router. Initially the ZigBee router would connected to the predefined ZigBee network having predefined PAN ID of the nearby railway station.
2. This module will be in turn connected to the level crossing gate control system
3. This module will monitor all the environmental conditions and send data to Azure cloud via railway station controller.
4. It will also take part in ZigBee routing to form a mesh network as discussed in previous section.
5. The signal post module and the level crossing gateway are placed 1 KM apart from each other. If the train approaching the signal module, the level crossing gate would be closed with all necessary audio-visual alerts.
6. The gate would remain close until the train has departure and is 1 KM away from the signal module.
7. After that the signal post would display green signal.

```
Train on wrong Track!!!
Train with ID: 40B7A183 is not allowed to run on track ID 1000696FFFFF.
 Alerted  the train driver regarding wrong track by blinking RED LED
Train on wrong Track!!!
Train with ID: 40B7A183 is not allowed to run on track ID 1000696FFFFF.
 Alerted  the train driver regarding wrong track by blinking RED LED
***************************************************************************
------------------------MESSAGE FROM TRAIN 40B7A183------------------------
1. Communication Link established ... Train is entered into zigbee network and is configured as Zigbee router
 2. Data is sent from train ID 40B7A183
--------------------END OFMESSAGE FROM TRAIN 40B7A183--------------------
***************************************************************************
***************************************************************************
------------------------MESSAGE FROM TRAIN 40B7A183------------------------
1. Train with ID: 40B7A183 is allowed to run on track ID 100069697F6F
2. HAPPY JORNEY!!!!!!!!!!!
--------------------END OFMESSAGE FROM TRAIN 40B7A183--------------------
***************************************************************************
***************************************************************************
```

Fig. 9: Output received from microcontroller after reading RFID Tag

Fig. 10: Train module along with RFID reader and Tags images

V. IMPLEMENTATION OF BACKEND

We are leveraging Azure cloud service as the backend for our system. We have inventory of all the assets deployed in our system. Below figure shows the schema of the database holding information of track IDs. Below is the schema of the RFID card and its events stored in Azure SQL table.

CREATE TABLE `cardevents` (`deptcode` tinytext NULL, `cardid` varchar(16) NULL DEFAULT NULL, `event` int(32) NOT NULL AUTO_INCREMENT, `servertime` timestamp NULL DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY (`event`) , INDEX `fk_accessevent_card_1` (`cardid`)) ENGINE=InnoDB AUTO_INCREMENT=2416 ;

CREATE TABLE `cards` (`cardid` varchar(8) NOT NULL, `allocated` tinyint(4) NOT NULL DEFAULT 0, `start` datetime NULL DEFAULT NULL, `end` datetime NULL DEFAULT NULL, PRIMARY KEY (`cardid`) , UNIQUE INDEX `carddex` (`cardid`)) ENGINE=InnoDB ;

All the railway control station uploads the information to Azure cloud bus [7] regarding sensors data and real time data. Azure in the backend analysis the data in real time and sends back data to the relevant control station if collision like situation is arises.

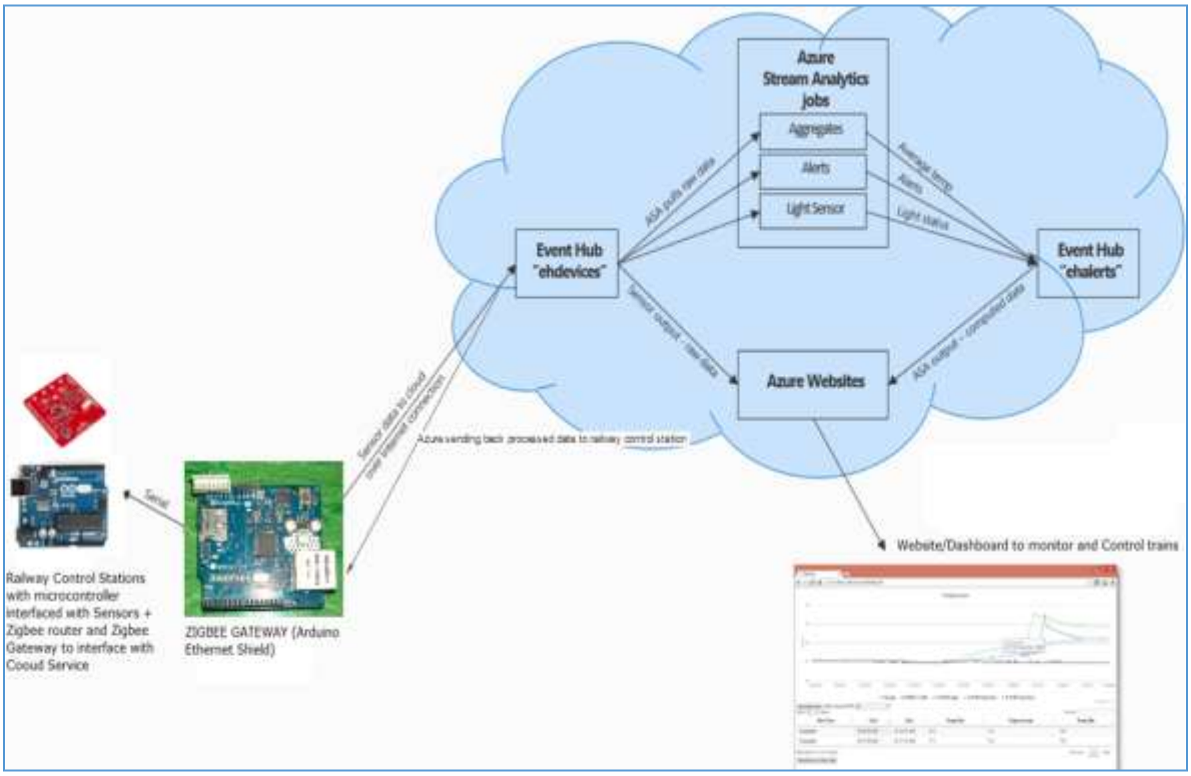

Fig. 11: ZigBee coordinator interface with Azure Cloud

Fig 8 shows the interface of ZigBee gateway (Arduino Ethernet Shield) with Azure event hub. Azure stream line analytics jobs processes all the real time data received to it via Event hub from railway control stations and environmental sensors. After analysis the Azure stream line analytics jobs necessary events back to event hubs which are sent to railway control stations and then propagated to individual trains via ZigBee network. All the real time data streams can be viewed with HTML5 capable browser.

5.1.1 Simulation of the Events and Azure Event Bus to Test Latency to Process Real Time Events

We have registered one endpoint (railway control station) on Azure service bus. We are inserting data into Azure queue using Windows phone. Windows phone is connected to Arduino on Pin 8 serial RX. Arduino is connected to Ethernet Shield having one static IP and a default gateway to internet router. Thus Windows phone sends event to Azure event bus and azure analytics in turns processes the queue and sends processed event back to event hub to the led. This setup was done to test the latency. In this setup we used Asynchronous Messaging Queue protocol to transfer messages to event hubs rather than HTTP as HTTP is very inefficient to carry messages as a payload. Figure 9 shows the simulation setup along with the dashboard to monitor real time events.

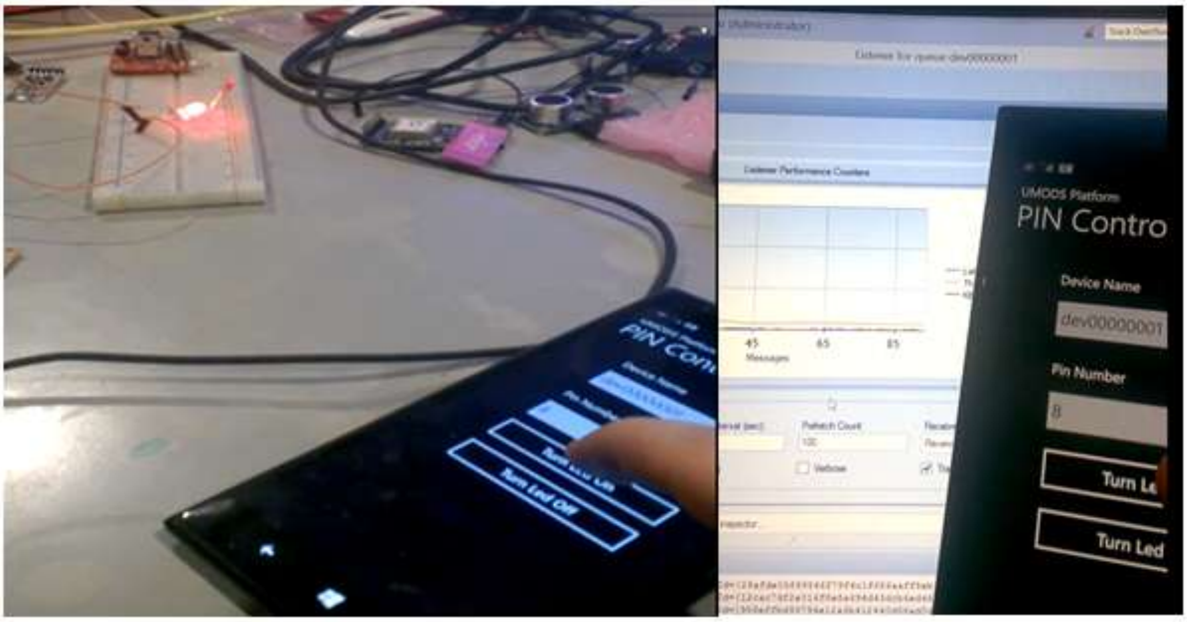

Fig. 12: Simulation of Signal Post module using Azure services

Fig. 13: Query run at Azure streamline Analytics to get real time data of temperature from Signal post module

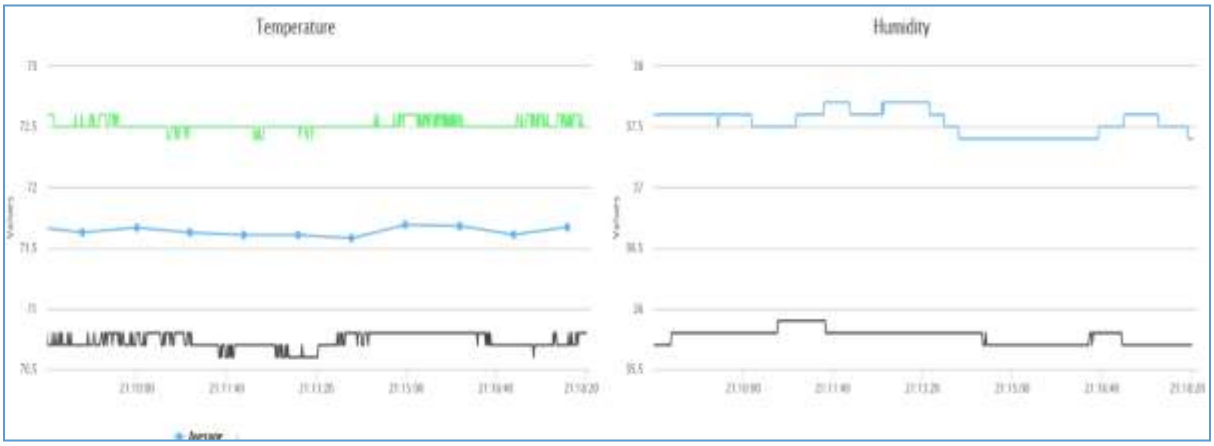

Fig. 14: Dashboard output for monitoring climatic conditions at train signal post

VI. EXPERIMENTAL RESULTS

6.1 Signal Strength Vs Distance

This set of measurement were carried out using the ZigBee module used in the implementation of this system. Approximately 150 measurements were done iterating for every selected distance.

Distance	1m	2m	3m	4m	5m	7.5m	10m	15m	20m
RSSI	-58.889	-60.732	-66.086	-66.577	-69.6	-69.474	-76.439	-77.685	-93.141
St.Deviat.	2.233	1.407	1.313	0.632	1.035	1.621	2.024	1.453	1.408

Table1: Distance, average RSSI [dBm] and standard deviation forZigBee module

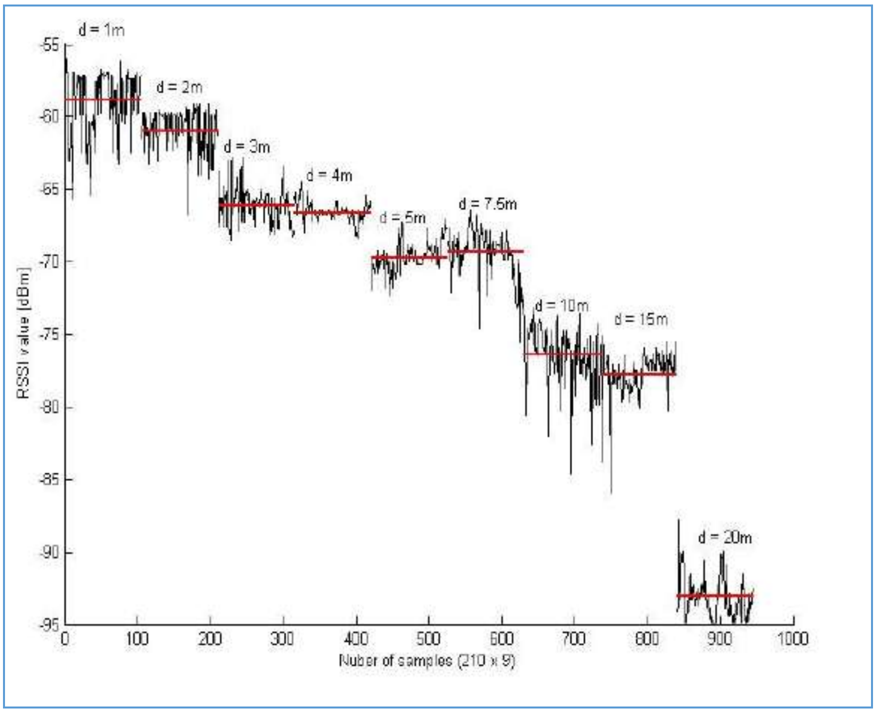

Fig. 15: Dashboard Output for Monitoring Climatic Conditions at Train Signal Post

Measurements were done in a real environment, not in a noise-free laboratory. This is because we wanted to place the modules within real working environment in which they would be normally be operating when deployed.

6.2 Zigbee Network Self-Healing Analysis

One of the most important characteristics of ZigBee is its self-healing capacity through mesh routing. So we tested the time cost of mesh routing by measuring the elapsed time between the elimination of one path and the search and creation of another. The test set up uses a diamond topology network as the one shown in Fig. 13.To perform the test we first start sending messages from ZR3 to ZC through ZR1, after some time we turn off ZR1 so that ZR3 needs to find another path to deliver the messages to ZC. The path must be found through ZR2, and we measure the time between the last message sent through ZR1 and the first sent through ZR2, this is what we called mesh routing recovery time.The results of this test are shown in Fig. 14 and there we can notice a maximum delay of 126 ms to find the new path when using a 20 ms transmission period. We can also notice the delay gets smaller when the transmit period between the messages is bigger than 20 ms; probably the bigger period permits the node to focus its resources such as memory and processing, to find the new path. The minimum delay we found was 85 ms. When we used maximum payload messages for the tests, with transmit periods under 20 ms it was not possible for the node to find the new path probably for the lack of memory to perform the search. Though for the 20, 40 and 50 ms tests, the node found the new path in less than 90 ms.

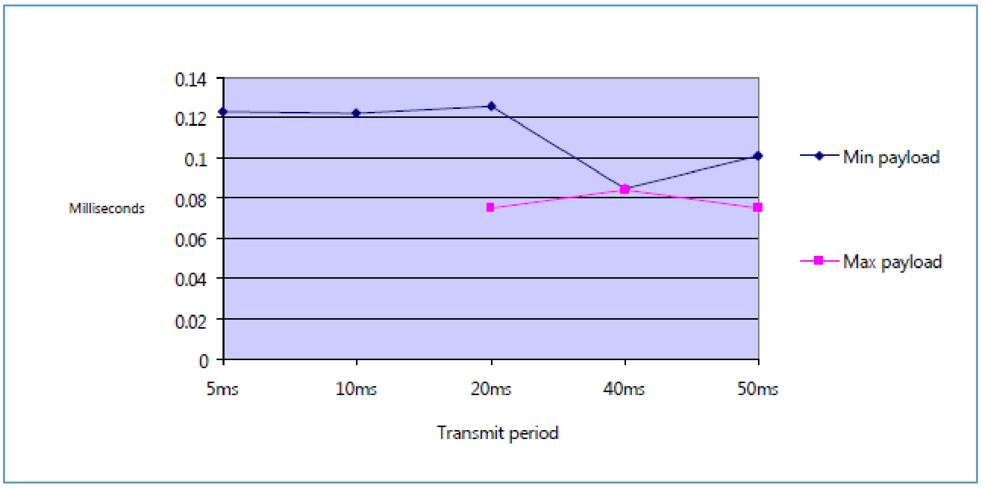

Fig. 16: ZigBee Mesh network throughput

6.3 Summary of the Results

Sr No	Parameter	Network	Theoretical	Practical
1	Data rate	ZigBee network	250Kbps	8.3Kbps
2	Nodes	ZigBee network	65000	1650
3	Retransmission time	ZigBee network	1msec	17.2msec
4	Power consumption		NA	0.035706 W when transferring 24 bytes of data
5	Latency	ZigBee network	NA	20msec
6	Battery life	ZigBee network	NA	1000 hrs at 34 bytes/second
7	Azure latency	Internet	NA	1 second
8	Time to join	ZigBee network	2 sec	8 seconds
9	Time to detect collision like situation	ZigBee network + Azure Analytics latency + Internet latency	3 sec	7 to 9 seconds

Table 2: Experiment Results Summary

VII. CONCLUSION

Collision avoidance systems are especially useful in bad weather conditions. In this project a design for automatically alerting train collisions and accidents at level crossing gate have been designed, simulated and tested.

It uses the advanced features of ARM and Arduino micro controller with RFID and ZigBee communication technique along with scalable Azure backend infrastructure, proves to be very cost effective, scalable, practically feasible without requirement of doing major infrastructure changes. From results we can see that ZigBee consumes very less power hence perfect candidate in embedded systems.Saving human life, protection against accidents and the communicable electronic systems are the salient features and the added advantage of this research.

REFERENCES

[1] Siror, J.K. ; Compute. Sci. & Eng. Dept., Shanghai JiaoTongUniv.,Shanghai,China; ShengHuanye ; Wang Dong ; Wu Jie, Use of Rfid Based Real Time Location Tracking System to Curb Diversion of Transit Goods in East Africa.

[2] Bhatt, Ajaykumar A, ‘An Anti-Collision Device Network – A train Collision Prevention System (TCPS)’.

[3] Z. Alliance, ZigBee Specification, 1st ed., ZigBee Standards Organization, http://www.zigbee.org, June 2005.

[4] M. Petrova, J. Riihijärvi, P. Mähönen, and S. Labella, “Performance study of IEEE 802.15.4 using measurements and simulations,” in Proc. IEEE Wireless Communications and Networking Conference (WCNC 2006), Las Vegas, NV, USA, vol. 1, April 2006, pp. 487–492.

[5] How to Create and Configure an Azure SQL Database http://azure.microsoft.com/en-us/documentation/articles/sql-database-create-configure/

[6] AMQP 1.0 support in Service Bus http://azure.microsoft.com/en-us/documentation/articles/service-bus-amqp-overview/

[7] Microsoft Service Bus and Access Control for Windows Azure platform AppFabric :http://www.microsoft.com/windowsazure/whitepapers/

Biographical Notes

Mr. Sahil Dange is presently pursuing M. Tech. final year in Electronics and telecommunication Engineering Department (Specialization in Embedded and VLSI systems) from Pune University GenbaMoze College of Engineering India guided by **Prof. A.D.Jadhav**