TMCAS:An MQTT based Collision Avoidance System for Railway networks

BiswajeebanMishra[1]
[1]Department of Software Engineering
6720 Szeged, Dugonics tr 13, Hungary
*Correspondence: mishra@inf.u-szeged.hu

***Abstract*—Rail is a convenient, economical, and widely used transport system for transferring passengers and goods all around the world. However, 'Safety' has been one of the most significant concerns of Railway Systems. With the tremendous improvement in the IoT, signaling system, and tracking technologies, the number of train accidents has lowered down over the past few years; still, the numbers are on an alarming level. Most of the current anti-collision systems use the fusion of RFID data, GPS sensor, Inertial Sensors, odometers, etc. to detect the exact position of a train on the track networks and get this data exchanged between the trains and transmitted to the central control room for necessary actions. In this paper, we propose an anti-collision system named "TMCAS" (Train Monitoring and Collision Avoidance System) that uses MQTT protocol to share and exchange position, speed, the signal at danger information between trains and central control room. TMCAS aims to automate train operations and act as a safety enhancement system by ensuring speed control, rear-end and head-on collision avoidance, and signal-at-danger avoidance. The concept and algorithm presented in this paper can be modified to fit into different real-time implementation needs or use cases.**

Keywords—Rail Network, Internet of Things, MQTT, MQTT Brokers, Anti-Collision

I. INTRODUCTION

Although safety is paramount for railway networks, many accidents and collisions occur due to human error, [6] such as exceeding the speed limit; signal passed at danger, faulty information sharing and wrong signaling and weather conditions, etc. The principal objective of this proposed system is to prevent collisions by breaking and stopping trains automatically if they run on the same track and approach towards each other. The system can also be implemented to avoid accidents at unmanned railway crossings. In anticipation of danger, TMCAS issues alert messages to loco-pilots to take necessary actions before it stops the trains automatically. The system is comprised of integrated with onboard RFID Reader [14], Microcontroller, GPS receiver, Intertial sensors, GSM module, RF transceiver. The essense of the proposed system is that, it deploys MQTT [3] protocol for reliable information exchange between trains and the central control room.

The following section of this paper discusses why MQTT protocol is chosen as an information career and what are its advantages over other available protocols.Section III summarises the overall functionality of the proposed system Section IV presentes a simulator named "TMCAS" as a proof of concept(POC)of the ideas presented in this paper and demonstrates its functionalities.

Finally, the paper is concluded in section V with a brief discussion about the future work.

II. WHYMQTT?

In this Internet of Things (IoT) era, millions of sensors, smart devices use the existing Internet Platform to talk to each other and share data. IoT networks rely on many different types of radio technologies (RFID, WLAN, WMAN, WPAN, etc.) [1],[2],[10] for communication. No matter what technology is used for machine to machine communication, all end devices must make their data available to the Internet to be used to derive useful information out of it. Here comes the role of M2M messaging protocols like MQTT, CoAP, AMQP, and HTTP [4],[7],[9] Out of these four widely accepted protocols, MQTT (MQ Telemetry Transport) [2] stands tall in various performance measures such as (Message Size vs. Message Overhead, Power Consumption vs. Resource Requirement, Reliability/QoS vs. Interoperability, Bandwidth vs. Latency, Security vs. Provisioning, M2M/IoT Usage vs. Standardization, etc.) [2],[8].

It was invented by Dr. Andy Stanford-Clark of IBM, and Arlen Nipper of Arcom, in 1999[3]. One advantage of MQTT over other protocols is that, it supports smallest and monitoring devices and transmit data over low-bandwidth or unreliable networks with very less consumption of power. Another big advantage is, being an open-source it can be customized to suit various communication needs for various transmitting environment. MQTT uses IANA registered port number 8883 for SSL/TSL connections, and 1883 port number for non-TLS connections. It is a publish/subscribe type protocol. It has three constituent components: Publisher, Broker and Subscriber. An MQTT client application that publishes messages to an MQTT server on a topic is called a Publisher. The MQTT Server is otherwise known as a Broker, and the MQTT client which subscribes to the topic to receive the published data is called a Subscriber. Around 45 thousand concurrent connected MQTT clients can be handled by an MQTT Broker 0.

A plaintext MQTT message has the least header size of 2 Bytes(fixed), along with an optional message-length header, and a message payload header[3] see Fig. 1. The latest version MQTT(V3.1) supports passing a username and password with an MQTT packet. Encryption across

978-1-5386-7214-3/18/$31.00 ©2018 IEEE

the network can be handled with SSL. An application that deploys MQTT can add additional security by encrypting the data it sends and receives.

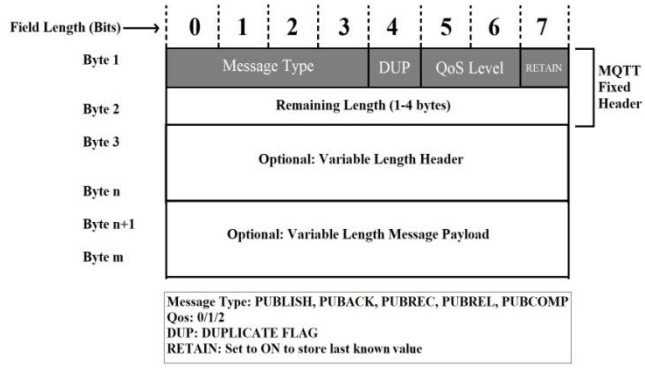

Fig. 1. TCP MQTT Packet Format

III. SUMMARY OF THE PROPOSED SYSTEM

A. Hardware Requirements

In this subsection we discuss about the hardware requirements for this proposed system see Fig. 2. In TMCAS every train is embodied with a Microcontroller (. i.e.: ARM Cortex A53), RFID Reader, GPS Receiver, Inertial sensors and GSM Module and RF Transceiver. Train tracks are divided into multiple distinct segments and numbered using passive RFID Tags that emit location of the tag, geometrical information of the track and possible speed restriction information, etc. [5], [11], [13]. The train equipped with RFID Reader reads the information stored on a RIFD tag and sends it to the computing unit for the further processing.

1) Passive RFID Tag

Passive RFID tags are cheaper and smaller. It doesn't require any external battery supply rather it uses the radio energy transmitted by the RFID Reader. It is usually used to transmit the exact location of the train and programmed to send the same information to every RFID reader that passes over it. RFID tags are placed on the railway track at a specific distance from each other.

2) RFID Reader

RFID readers are usually attached below the train engine to receive the transmitted information from the RFID tag and send them to the onboard computer for further processing.

3) GSMModule

GSM module or a GPRS module is a chip or circuit that is used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. GSM operates at different frequency bands from 850MHz to 900MHz and 1800MHz to 1900MHz. GSM-R is built on GSM technology, and benefits from the economies of scale of its GSM technology heritage. It aims at being a cost-efficient digital replacement for existing incompatible in-track cable and analog railway radio networks. It uses a lower extension of the 900MHz frequency: 876 MHz to 915 MHz for uplink and 921 MHz to 960 for downlink.

4) RF Transceiver

An RF Transceiver receives and transmits serial data as radio signals. The speed of transmission varies between 1Kbps to 10Kbps. The RF module often uses a pair of encoder/decoder at 434 MHz [12].

(As MQTTClient library supports a replaceable networking class such as MQTTEthernet and MQTTSocketetc. [3]. It can be customized to work with any networking interface. It is to be noted that the design principle of MQTT ensures functionality under unreliable and low bandwidth network environments. This proposed system primarily advocates to use GSM-R for communication; In case of failure of GSM-R for any reason, RF transceiver can be used to send and receive data over radio signals)

5) GPS Receiver

A GPS navigation device or GPS receiver is capable of receiving information from GPS satellites and then to calculate the device's geographical position. It can also be programmed to derive position and direction of an object on a map.

6) ARM Cortex-A53

The ARM Cortex-A53 implements ARMv8-A 64-bit instruction set. It is a superscalar processor, capable of dual-issuing some instructions. Cortex-A53 ensure higher level computing performance combined with fast interrupt handling at a very lower level of power consumption.

7) LDR Sensor:

LDR is a low cost and simple light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity. Light-dependent resistors are typically used at unmanned gates or crossings to detect the presence of any object or the absence of light[7].

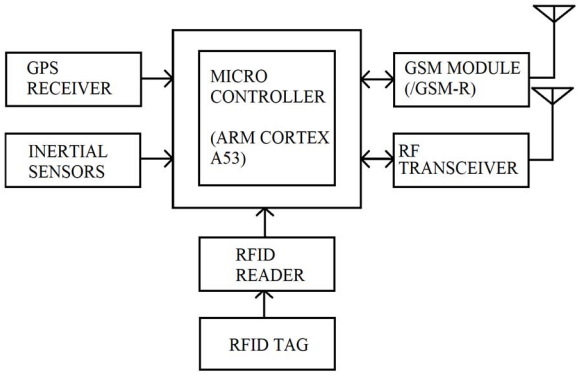

Fig. 2. Onboard Hardware Environment

B. Software Implementation

TMCAS comprises of three different software modules. See Fig. 4.

1) MQTT BASED TMCAS MODULE FOR TRAIN ENVIRONMENT (T-TMCAS)

This module is responsible for gathering, sorting and sending all the information about the running train(such as train number, speed, current position, driving direction) to the central control room via an MQTT Broker, receiving warning/alert messages, break and stop commands and signal related information from the central control room

and displays them all in real-time inside the cabin to the loco pilot for necessary action. It can also take prompt action when the loco-pilots are not able to control. It sends all the information to the Central control room through the GSM module or RF transmitter. T-TMCAS. Fig. 3 depicts how T-TMCAS module encapsulates train id, track id, speed data and direction data together as a location info packet and transmits or publishes it over the chosen transmission medium (GSM/RF transmitter) to the deployed MQTT broker. T-TMCAS modules installed on the trains can also be customized to support fail-safe mode to work independently with MQTT Broker Server in the event of communication failure between the broker and central control room.

2) MQTT BASED TMCAS MODULE FOR CONTROL ROOM ENVIRONMENT (CR-TMCAS)

This Module is the brain of entire TMCAS system. It runs at the central control room. It basically keeps track of all online and offline trains. It collects all the positional information of different running trains; issues commands through GSM-R or radio frequency transmission to automatically break and prevent a collision if two trains approach toward each other on the same track. It first reduces the speed of both the trains before they are put to a halt at a specified distance from each other. Exchange of information is managed by the deployment of MQTT and MQTT Broker. Upon detecting an object in an unmanned railway crossing, it can also send alert messages to the loco-pilot of the train approaching the crossing to take necessary action and or issue commands to slow down and stop the train automatically before it reaches the unmanned railway crossing.

3) MQTT BASED TMCAS MODULE FOR RAILWAY CROSSING (RC-TMCAS)

This module deals with collision avoidance at unmanned railway crossings. This module works with the following hardware set up; LDR sensor, GSM Module, and ARM Cortex microcontroller. LDR sensor continually detects an object on the crossing. If finds any object on the crossing, it informs the CR-TMCAS module for the necessary action.

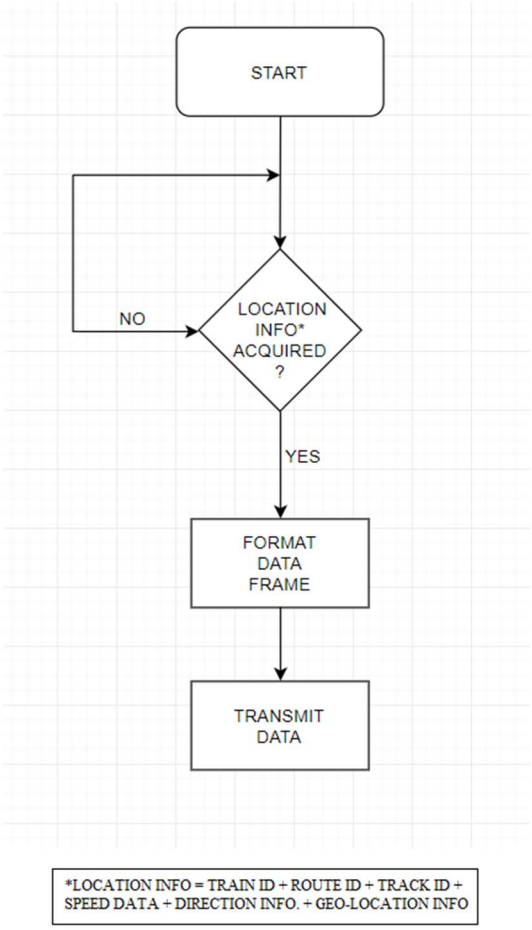

Fig. 3. Data Flow Chart of T-TMCAS Module

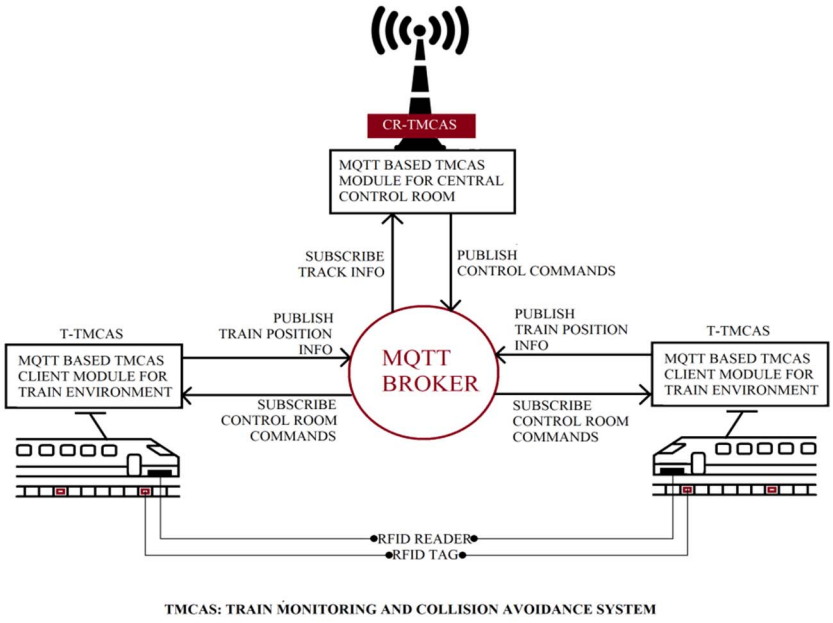

Fig. 4. Overview of TMCAS System

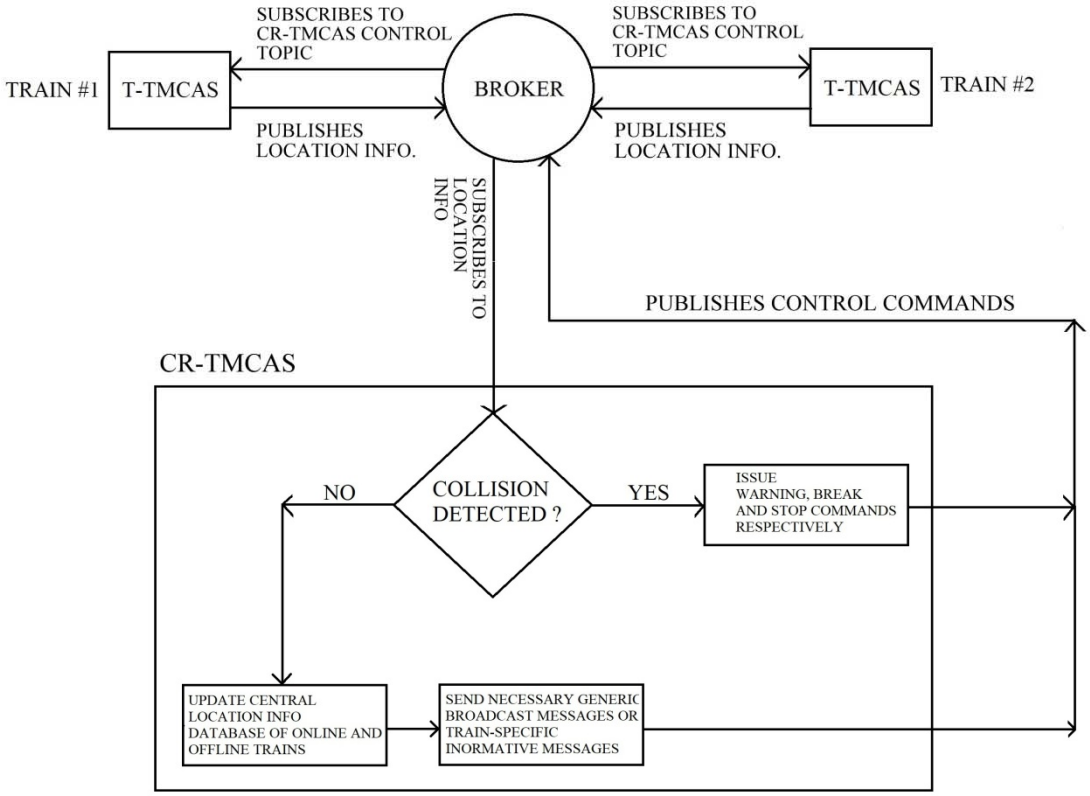

Fig. 5. Workflow diagram of TMCAS Simulator

IV. TMCAS

Train Monitoring and Collision Avoidance System or in short "TMCAS" is a simulator that embodies the concept of MQTT based anti-collision system presented in this paper. We have released two different binaries of TMCAS on GitHub. One is a C binary for Linux environment and the other one is a .NET binary for Windows environment. For the demonstration purpose we have built this simulator to work with publicly accessible Broker servers [17]. In this version of binaries, we address these three types of collision scenarios-

- Head-on-Collision of trains running in opposite direction to each other on the same track with same speed
- Head-on-Collision of trains running in the same direction on the same track but with different speed.
- Head-on or rear-end collision when one train is static at a position and other train moves toward it.

In TMCAS, we have divided the entire network of rails into7 routes where each route has 15 track segments or cells (15 track ids). Itworks with public MQTT broker servers to simulate the collision scenarios and shows the interaction of T-TMCAS and CR-TMCAS modules of proposed TMCAS system. All the settings of the simulator are configurable see Fig. 6., i.e. a user can set name of the public broker [17] he or she wants to use, source and destination stations, speed, direction of the trains, and the distance (number of cell- distance needed) to issue break and stop commands from the CR-TMCAS module can be manually configured. See Fig. 6.

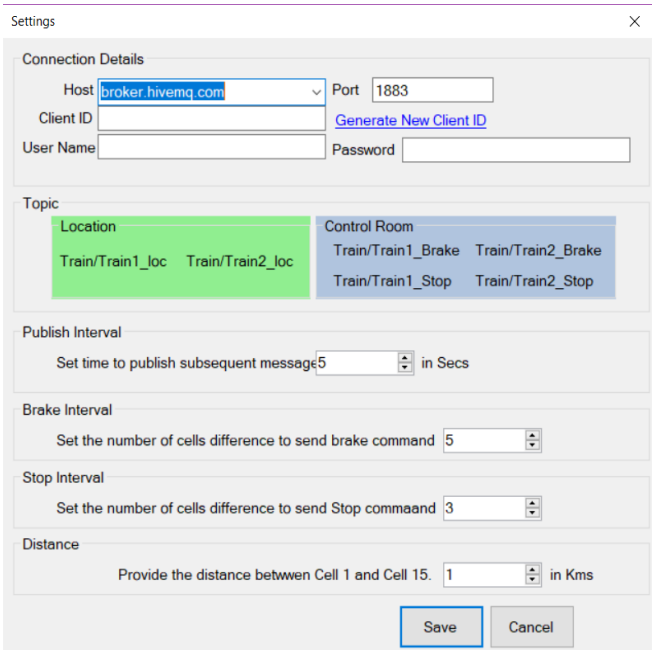

Fig. 6. General Settings of TMCAS Simulator

Fig. 7. Shows the screenshot of the TMCAS simulator before collision

Fig. 8. Shows the screenshot of the TMCAS simulator after collision being avoided by stopping the trains before they reach the possible point of collision

For example, when a user sets 5 cells-distance and 4 cells-distance see Fig. 7 and Fig. 8 to issue break and stop commands respectively from the central control room (CR-TMCAS module), the trains heading opposite to each other on the same track will be braked and stopped in 5 and 4 cells-distance respectively from the possible point of head-on-collision. In the current implementation of the simulator the central control room i.e. CR-TMCAS module divides the distance between the starting points of trains by 2 and passes resultant value through an abs(absolute) function to obtain the possible point of collision of trains running on the same route. When trains start running on the track the T-TMCAS modules installed on the trains keep on publishing positional information of the running trains to a

MQTT broker server on train-specific unique topic heads and simultaneously subscribe to the control topics of central control room module (CR-TMCAS) to receive alert, control commands from it. The CR-TMCAS module subscribes to location information topic/s and analyses the geo-location, direction, speed information of different trains. In anticipation of collision, CR-TMCAS issues alert, break and stop commands etc. to the danger-approaching train or trains. Upon receiving the alert, break and stop commands from the CR-TMCAS module, the T-TMCAS module acts promptly to slow the train/s down and finally stops the train/trains automatically before they could hit a possible collision See Fig. 8. Fig. 5 represents the workflow diagram of the TMCAS simulator.

V. CONCLUSION

Implementation of Collision Avoidance Systems can give a big boost to railway safety. In this paper, we have discussed an approach for building an anti-collision system for rail networks using MQTT protocol as its central backbone and presented a simulator named "TMCAS" as a proof of concept. However, it is not end of the road. The Real time implementation may vary and involve a lot of challenges. Hence, the proposed concept and algorithm may be modified to fit into different real-time implementation needs or use cases.

ACKNOWLEDGEMENTS

The author would like to show his heartfelt gratitude to his colleagues Attila Kertész, Tibor Gyimóthy and especially to Biswaranjan Mishra, Wind River Systems, Bangalore and Adhyatmananda Pati, Revenue and Disaster Department, Govt. Of Odisha, India for their constructive comments, suggestions and guidance that made this research work possible.

REFERENCES

[1] V. Lampkin et al., Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ, Telemetry, IBM Redbooks publication, 268 pages, 2012.

[2] Nitin Naik, Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQPand HTTP. IEEE International Systems Engineering Symposium, Vienna, pp. 1-7, 2017.

[3] MQTT Version 3.1.1, October 28, 2018 (accessed April 04, 2018),http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html.

[4] Paolo Bellavista, Alessandro Zanni, Towards Better Scalability for IoT-Cloud Interactionsvia Combined Exploitation of MQTT and CoAP, IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI),2016.

[5] Miltos Kyriakidis, Kam To Pak, Arnab MajumdarArnab Majumdar,
Railway Accidents Caused by Human Error,Transportation Research Record Journal of the Transportation Research Board, October 2015.

[6] R.Immanuel Rajkumara P.E.Sankaranarayananb G.Sundaric,An approach to Avoiding Train Collision in Railway Sectorsusing Multi Agent System,3rd International Conference on Recent Trends in Computing, 2015.

[7] S. Bandyopadhyay and A. Bhattacharyya, "Lightweight internet protocolsfor web enablement of sensors using constrained gateway devices,"in Computing, Networking and Communications (ICNC), 2013 InternationalConference on. IEEE, 2013, pp. 334–340.

[8] N. Naik, P. Jenkins, P. Davies, and D. Newell, "Native web communicationprotocols and their effects on the performance of web servicesand systems," in 16th IEEE International Conference on Computer andInformation Technology (CIT). IEEE, 2016, pp. 219–225.

[9] A. Foster, "Messaging technologies for the industrial internet and the
internet of things whitepaper," PrismTech, 2015.

[10] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-
Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp.
11–17, 2015.

[11] Smita S. Bhavsar, Prof. A.N. Kulkarni,Train Collision Avoidance System by Using RFID, IEEEInternational Conference on Computing, Analytics and Security Trends (CAST), 2016.

[12] Kurhe Jyoti, Gophane Prajakta, Kadam Madhuri, Panchal, Anubha "Train Collision Detection and Avoidance" International Journal of Engineering Science and Computing, March 2016.

[13] K.Govindaraju, F.Parvez Ahmed, S.Thulasi Ram, T.Devika" A Novel
Approach Of TrainPrevention System From CollisionUsing AvrMicrocontroller" International Journal Of Innovative Research InElectrical,Electronics,Instrumentation &Control EngineeringVol.2,Issue
February 2014.

[14] Nayan Jeevagan, Pallavi Santosh , Rishabh Berlia , Shubham Kandoi
"RFID Based Vehicle Identification During Collisions " IEEE 2014 Global Humanitarian Technology Conference Gate Protection System by Konkan Railway.

[15] "Nist sp 800-132, recommendation for password-based key derivation part1: Storage applications," http://nvlpubs.nist.gov/nistpubs/Legacy/SP/
nistspecialpublication800-132.pdf, (accessed on March04, 2018)

[16] TMCAS Version 1.1, May 6, 2018 (accesed May 6, 2018) https://github.com/Biswajeeban/TMCAS

[17] List of Publicly accesible MQTT Brokers, (accesed May 6, 2018), https://github.com/mqtt/mqtt.github.io/wiki/public_brokers