ICTO 2017: Information and Communication Technologies in Organizations and Society

Paris, March 16th and 17th, 2016

A PERVASIVE IOT SCHEME TO VEHICLE OVERSPEED DETECTION AND REPORTING USING MQTT PROTOCOL

Complete Research

Abstract

One particular concern that Public Safety Organization (PSO) must account for is the excess of speed of vehicles in motion. The high speed is typically responsible for a significant proportion of the mortality and morbidity that result from road crashes. According to the World Health Organization (WHO), "an increase in average speed of 1 km/h generally results in a 3% higher risk of a crash involving injury, with a 4–5% increase for crashes that result in fatalities"[1]. Various ineffective proposed methods and solutions have been implemented to control speed limits; for instance, Speed Detection Camera System (SDCS), Radio Detection and Ranging (RADAR), Light Detection and Ranging (LIDAR). This paper conveys an innovative, pervasive, effective and adaptable Internet of Things (IoT) system to detect and report vehicle overspeed as well as issuing tickets and fines. Our aggregated prototype is composed of five components: IoT vehicle on-board unit, Message Queuing Telemetry Transport (MQTT) broker, application logic server, data storage, and monitoring engine. A software simulation has been implemented and tested as a proof of concept. This novel technique is restricted only for governmental use since it surrogates the contemporary aforementioned speed detection systems paving the way toward a smarter and sustainable solution, and thus ensuring public safety.

Keywords: PSO, WHO, vehicle, overspeed, IoT, Radar, MQTT, ticket.

1 Introduction

Many reasons lead to fatal road crashes that occur disproportionally and are stretched from road state, car situation, weather condition, driver alertness to speeding which remains a major safety concern on the nations' roadways. Vehicle high speed is considered a crucial factor that typically leads to a significant increase in the morbidity and mortality rate. Moreover, governments and PSOs invest enormous amount of money and human resources to provide efficient traffic surveillance systems which control the excess of speed and hence enforcing traffic speed laws. The (WHO) organization states that road traffic crashes are predicted to become the 7th leading cause of death by 2030. An adult pedestrian's risk of dying is less than 20% if struck by a car at 50 km/h and almost 60% if hit at 80 km/h. A 30 km/h speed zones can reduce the risk of a crash and are recommended in areas where vulnerable road users are common like residential and schools areas. According to the Association for Safe International Road Travel (ASIRT), nearly 1.3 million people die in road crashes each year, on average 3,287 deaths a day. Road crashes resulting from high speed cost $518 billion globally, constituting 1-2% of individual countries' annual Gross domestic product (GDP) [2].
Many speed devices are currently available: The RADAR, the LIDAR, and the Speed Detection Camera System (SDCS). However, the use of these speed control devices has not resulted in definitive conclusions about their effectiveness as stated in the next section.

Based on the above aforesaid facts, an ample and doable solution for high speed detection becomes a must. Hence, the research question rises up: "How can we provide an innovative, faultless, and effective scheme for overspeed detection?"

To combat the speeding problem, we propose an effective and innovative IoT scheme which depends on the following components: a software component, for instance, an application that relies on the MQTT lightweight protocol to report instantly the speed of the vehicle, and to issue a ticket when overspeed threshold value is defeated; an on-board unit integrated into the vehicles which is responsible for broadcasting the vehicle geolocation information as well as its speed to a processing server; a processing server to retrieve and calculate the speed limit taking into account several factors: state of the road, traffic congestion, weather conditions and the like.

The main contributions of this paper are: (a) Developing a new smart IoT solution which helps governmental authorities in supervising vehicle overspeed. (b) Issuing and collecting car tickets autonomously. (c) Implementing a cost efficient and accurate solution for overspeed road control. (d) Collecting geographical data which can be fed into a data mining engine to extract roads conditions. (e) Providing traffic descriptive statistics reports for some critical areas.

This paper starts with descriptive and inferential statistics about road traffic crashes, high death rates, prorated cost as well as its influence on the individual countries' annual GDP. Sections II, III, and IV, describe the related work, the proposed scheme, the design and implementation respectively. Sections V, VI, and VII expose results, performance, and conclusions and future work.

2 Related Work

This section discusses and studies the traditional radar systems, their requirements, functionalities and their intrinsic drawbacks.

2.1 RADAR (Radio Detection and Ranging)

There are two types of radar that are commonly used in almost all countries by law enforcement personnel to measure the speed of moving objects. Their functionalities are based on Doppler shifts to measure the speed of vehicle.

Fixed high way radar: It calculates vehicle's speed by means of sensors and capturing still images. This type of radar is considered extremely expensive. Its cost ranges from $20000-$30000 [3].

Mobile inner town radars or radar guns: This device may be hand-held or vehicle mounted. It calculates vehicle's speed by means of sensors, and it needs an operator to capture the images.

Various limitations of the radar system are perceived and not limited to: a) User training and certifications are required, b)Installation and deployment requires planning and mathematical consideration for better field of view, c) Radar can take up to 2 seconds to lock on and hence, cannot detect two excessive speeders simultaneously, d) Large targets close to radar can saturate or hide other smaller objects; therefore, the radar fails to locate the vehicle, e) Radar-triggered cameras are imperfect and can result in tickets being generated for false readings, f) Human intervention is required. They have no mechanism of sending the captured images. The authorities have to make periodic stops to collect the films

2.2 LIDAR (Light Detection and Ranging)

The Lidar system relies on the principles of time-of-flight of two or more short wave length laser pulses. Sweep for instance, is a $250 Lidar with range of only 40 meters [4]. If we calculate the number of Lidars in a country, their cost will be very high.

Some of the disadvantages are: a) Particles (dust, water) in air can limit range, b) Rounded surfaces, the color black, blue, and violet are poor reflectors, c) Alignment can cause severe error, d) Extreme sunlight can be damaging.

Other technologies can be used to avoid and defeat radar and Lidar systems:

- Laser detectors and radar detectors. They detect if the speed is being monitored and warn the driver.
- Laser jammers and radar jammers. They jam the laser and the radar signal and return a scrambled signal so as the radar speed camera cannot process.[5]

2.3 SDCS (Speed Detection Camera System)

SDCS is a camera that uses image processing to detect traffic regulation violations. It can be mounted beside or over a road or installed in an enforcement vehicle.

Some of the perceived drawbacks are: a) SDCS method requires large database for storing video, therefore the cost of this method is higher than of the RADAR and LIDAR, b) In December, 2012, Speed Camera Contractor Xerox Corporation admitted that cameras they had deployed in Baltimore city were producing erroneous speed readings, and that 1 out of every 20 citations issued at some locations were due to errors c) One issue is the potential conflict of interest when private contractors are paid a commission based on the number of tickets they are able to issue.
The purpose of the speed camera program is to improve safety by reducing unsafe speed and must be persuasively and evidently communicated to the public. As a matter of facts, signs announcing the possible presence of speed cameras should be obviously posted throughout the enforcement area. To do otherwise, the suspicion that those cameras are being used mainly for revenue purposes rather than safety reasons [6].

Based on the above discussion, these traditional speed devices need to be replaced by an automated system having better precise outputs, less expensive, and exclude human factor.

3 The Proposed Scheme

This section outlines our proposed system at a high level scope. The system is composed of the following phases: (a) Service's registration and IoT device integration, (b) Speed and geo-location reporting (c) Overspeed detection and tickets' issuing.

3.1 Service's registration and IoT device integration

The IoT device must be integrated into every vehicle through the on-board Diagnostic (OBDII) plugin standard. After installation, this device automatically acquires the VIN (Vehicle Identity Number) and maps it to its Universal Unique Identifier (UUID). The combination of the VIN and the UUID ensures authentication and genuineness and thus, prevents device's counterfeiting. The first time the device is plugged in into the moving object, its mapping record is then posted to the server's database for registration and consequently, for post-matching. Since the database is managed by the government, then there exist a mapping among the three attributes VIN, plate number, and the vehicle's owner. If the IoT device has been plugged out intentionally or unintentionally then the server can detect the action as described technically in section 4.3 and consequently, the government applies the appropriate measures.

3.2 Speed and geo-location reporting

The IoT device reports repeatedly the speed of the car including its geo-location for a configurable time interval. In order to use the full potential of IoT paradigm, the device reports the data to the main server using the lightweight protocol MQTT over the mobile network. MQTT is an open-source protocol for passing messages between multiple clients through a central broker. The MQTT architecture is broker based, and uses long-lived outgoing TCP connection to the broker. MQTT can be used for two way communications over unreliable networks. It is also compatible with lower consumption devices [7]. In our system, each MQTT message is composed of the vehicle's speed, its geo- location, its VIN, and the date & time. In case the TCP connection is disrupted then all MQTT messages are stored temporarily on a secondary storage device. When the IoT device is reconnected, the stored messages are republished again to the server. This ensures service availability at any time.

3.3 Overspeed detection and tickets' issuing

To be able to issue overspeed ticket, there should be a mechanism to map the vehicle's geo-location to the exact street name or number. Many such services are available in the cloud through well-defined Application Programming Interface (API). But due to the high number of expected transactions published constantly to the server, a bandwidth problem might arise. Therefore, a dedicated server is developed to perform reverse geo-location offline.

Two alternative models have been proposed for overspeed detection: the static model and the adaptable model. The first is implemented by comparing the current vehicle's speed and its associated attributes to a speed record located into a database server provided by the government. The second is achieved through an adaptable solution based on weather forecast, and a universal standard adopted by many states and cities for establishing regulatory speed zones.

With respect to tickets' issuing, if the system detects overspeed then a vehicle record representing a ticket is added to a governmental database dedicated for that purpose (see section 4.3.1 for ticket's attributes). Consequently, the governmental database server maps the vehicle's VIN to its plate number and determines the fine that has to be paid by the car owner (owner of plate number).

Having described the system at a high level scope, the following section illustrates the system's design and describes its implementation in details.

4 Design and Implementation

4.1 System Architecture

Below is the system architecture of the conceptual model that exposes the model, the behavior, and the views of our proposed system in a sequence.

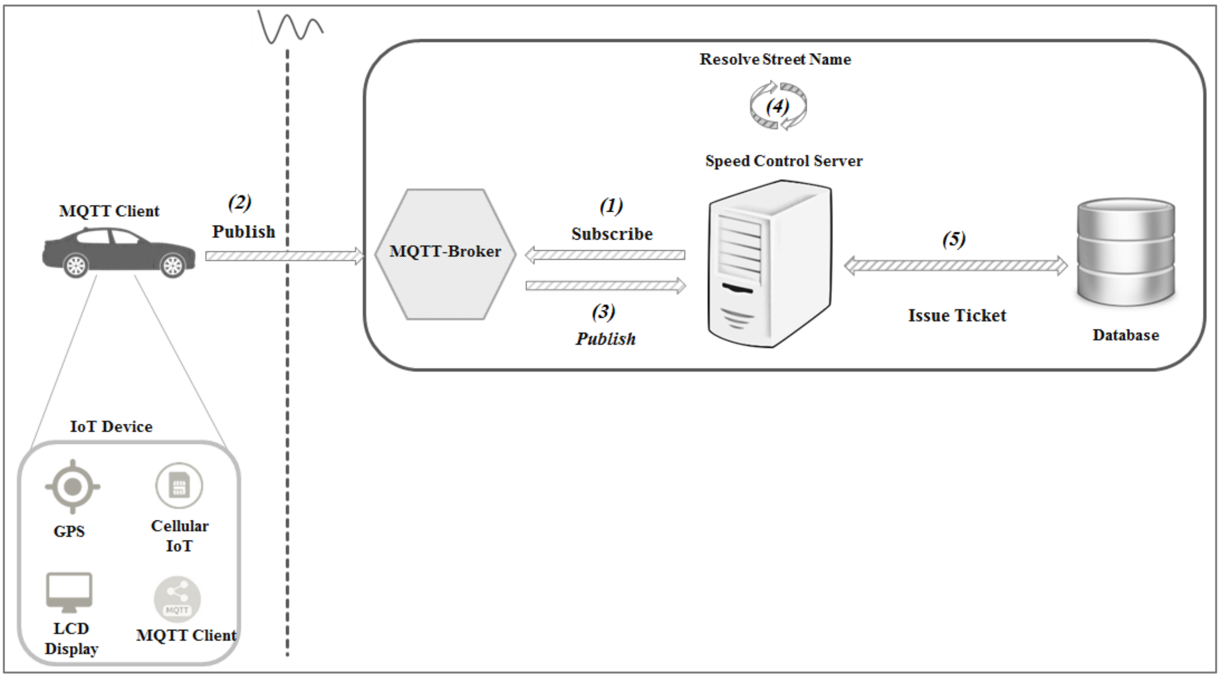

Figure 1. System Architecture

Figure 1 illustrates the consecutive steps of the different components of the system and how they interact with each other:

1. The speed control server which is considered as an MQTT client subscribes to the MQTT-Broker server to receive the messages published by all the moving vehicles.
2. The vehicle publishes its identity (VIN) as well as the speed, the GPS coordinates, and the time stamp.
3. Since the speed control server is already subscribed to the same published topic[1] (see Figure 2) of the vehicle; therefore, it receives the published information.
4. The speed control server is constantly performing reverse geo-location to map the received coordinates against the street name/number and hence, it detects the speed limit.
5. In case of overspeed detection, a ticket is issued spontaneously and stored in a database.

4.2 Device Components

4.2.1 Cellular IoT

Our IoT device uses the cellular 3G module through a Subscriber Identity Module (SIM) card to establish all kind of wireless communications from and to the server.

[1] A topic in MQTT technology represents the key that identifies the information channel to which payload data is published

It is required to implement cellular IoT 3rd Generation Partnership Project (3GPP) technologies: Extended coverage Global System for Mobile communication (ECGSM), Long Term Evolution (LTE), Long Term Evolution Machine to Machine LTE-M, and the new radio access technology Narrowband IoT (NB-IoT) specifically tailored to form an attractive solution for emerging low power wide area (LPWA) applications. [8]

4.2.2 Global Positioning System (GPS):

The GPS navigation is a component that accurately calculates geographical location by receiving information from GPS satellites. [9] The GPS device is used to send to server the exact vehicle location (longitude and latitude).

4.2.3 MQTT Client:

MQTT client API is installed on the device to enable the “publish/subscribe” model to the MQTT broker located on the server side as shown in Figure 2. Each vehicle’s published message includes a topic and its corresponding payload. The topic designates the routing information for the broker. Clients that subscribe to a specific topic receive the message pertaining to the topic’s key.

4.3 Software Design

Figure 2 depicts the software design of the system. On the server side, Mosquitto™ which is an open source MQTT message broker is adopted as the system’s broker. It is responsible to distribute the messages related to a topic to all its subscribers.

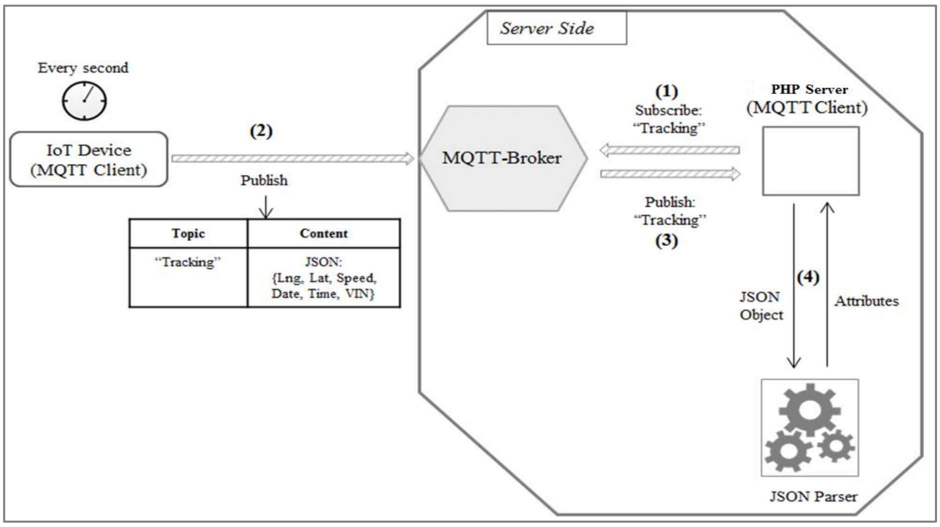

Figure 2. Software Design

In our software design, the vehicle equipped with the IoT device publishes to the topic “Tracking” a JSON object (the topic’s payload) containing the following parameters:

a. Longitude: East-West geo coordinate of a point on the earth's surface.
b. Latitude: North-South geo coordinate of a point on the earth's surface.
c. Speed: Using the GPS’s internal clock, the speed is calculated by measuring the time the vehicle needs to traverse between two points. As an alternative to GPS, the vehicle speed can also be obtained through the OBDII interface.
d. Date and Time: The National Marine Electronics Association (NMEA) data generated from the GPS is converted to readable format to extract the current date and time.
e. VIN: A unique combination of 17 letters and digits to identify a vehicle.

One of the subscribers to the MQTT broker is the speed control server which uses the Mosquitto-PHP library to become an MQTT client. Once the control server receives a message, it parses it retrieve every attributes separately.

Our software design is based on two alternatives: The first is the static model which relies on offline reverse geocoding for speed limit detection. The second is the adaptable model in which the speed limit parameter is detected dynamically as described in section 4.3.2. This method is based on the continuous weather forecast and the 85% percentile theorem.

4.3.1 Static model

The static model is illustrated in Figure 3 as shown below:

1- The PHP server sends the longitude and latitude parameters to a java application through PHP-JAVA-Bridge.

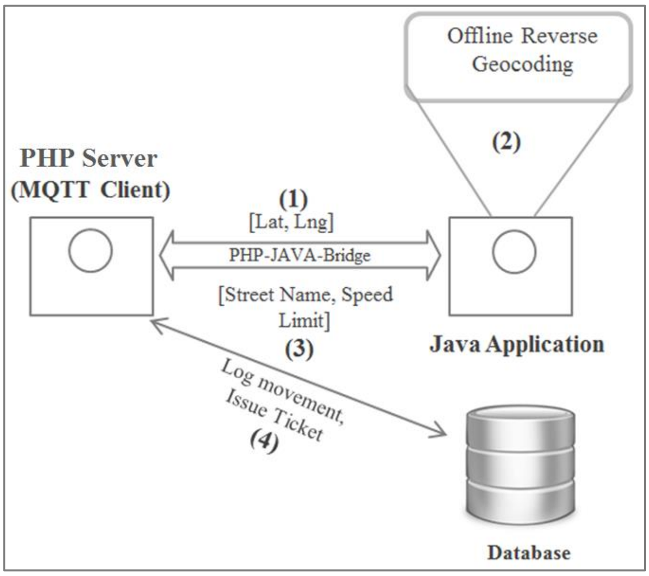

Figure 3. Static Pattern

2- KD-Tree [10] (Figure 4) for nearest neighbor lookup algorithm is implemented in Java to perform offline reverse geocoding. The Java application accepts as input the longitude and the latitude, and then compares them against a database provided by the government containing all country's street names and their corresponding speed limit.

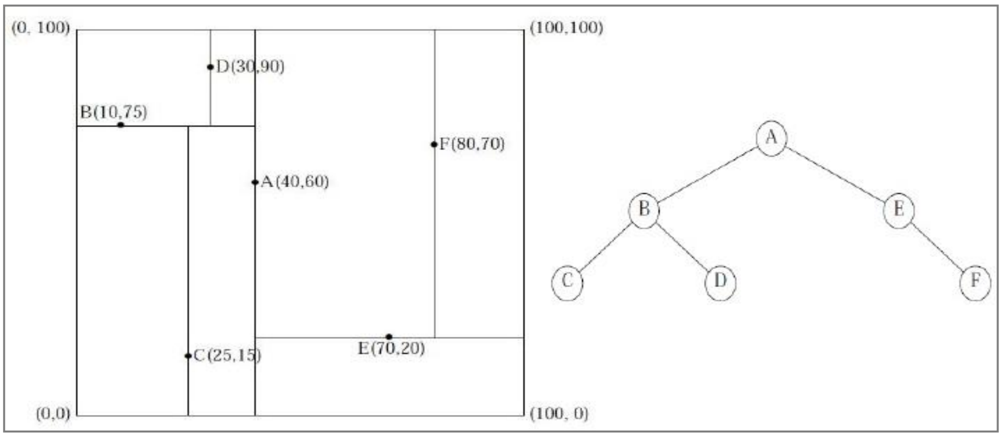

Figure 4. Spatial indexes – KD tree

3- The street name and the speed limit are returned to the PHP server for processing.
4- The PHP server extracts the speed from the JSON object and checks if it exceeds the speed limit. A ticket is issued and stored into the database in case of overspeed detection.

The ticket record's attributes are: VIN, speed limit, longitude and latitude, exceeded speed, date and time.

4.3.2 Adaptable model

In the adaptable model, the speed limit is considered as variant parameter and is calculated according to two factors: (1) 85th percentile theorem of vehicles speed, (2) continuous weather forecast.

85th percentile

The 85th percentile speed is the speed that 85 percent of vehicles do not exceed. Since the system is collecting all vehicles 'velocities as well as their locations, then, the 85th percentile theorem can be used to determine the new speed limit. Every ***T*** seconds/minutes (configured per country), the speed control server calculates the new speed limit per cluster and updates the database accordingly. A cluster as shown in Figure 6 is a segment defined by the country in a highway. The dynamic clustering system for highways is still an area under research.

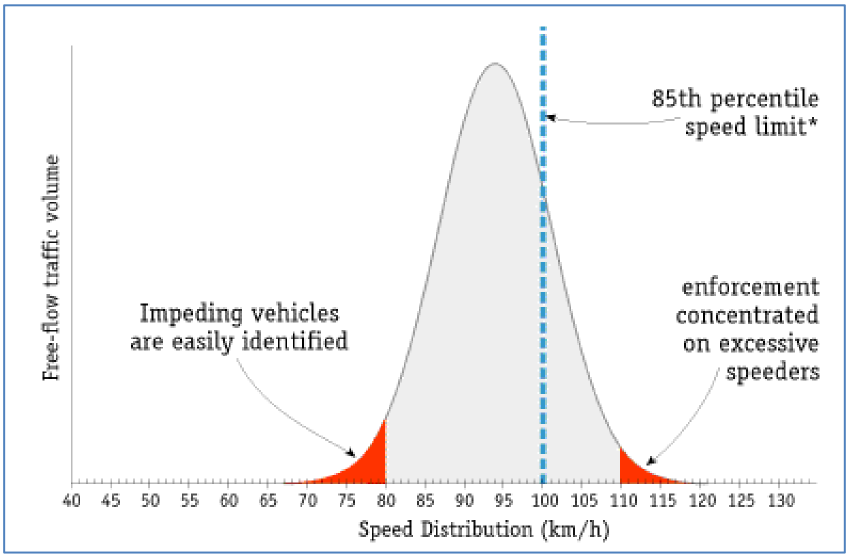

Figure 5. Adaptable Pattern (Source: http://michigandistilled.org)

The use of the 85th percentile speed concept is based on the theory that: 1) the large majority of drivers: Are reasonable and prudent, 2) do not want to have a crash, 3) desire to reach their destination in the shortest possible time.
A speed at or below which 85 percent of people drive at any given location under good weather and visibility conditions may be considered as the maximum safe speed for that location.[11]

How to determine if a car lies within a specific cluster?

To determine whether a car lies within a specific road cluster an algorithm proposed by Philippe Reverdy is used. By considering a road cluster as a polygon, this algorithm computes the sum of the angles made between the test point and each pair of points making up the polygon. If this sum is $\mathbf{2\pi}$ then the point is an interior point (vehicle lies within the cluster), if it is **0** then the point is an exterior point (vehicle lies outside the cluster). [12]

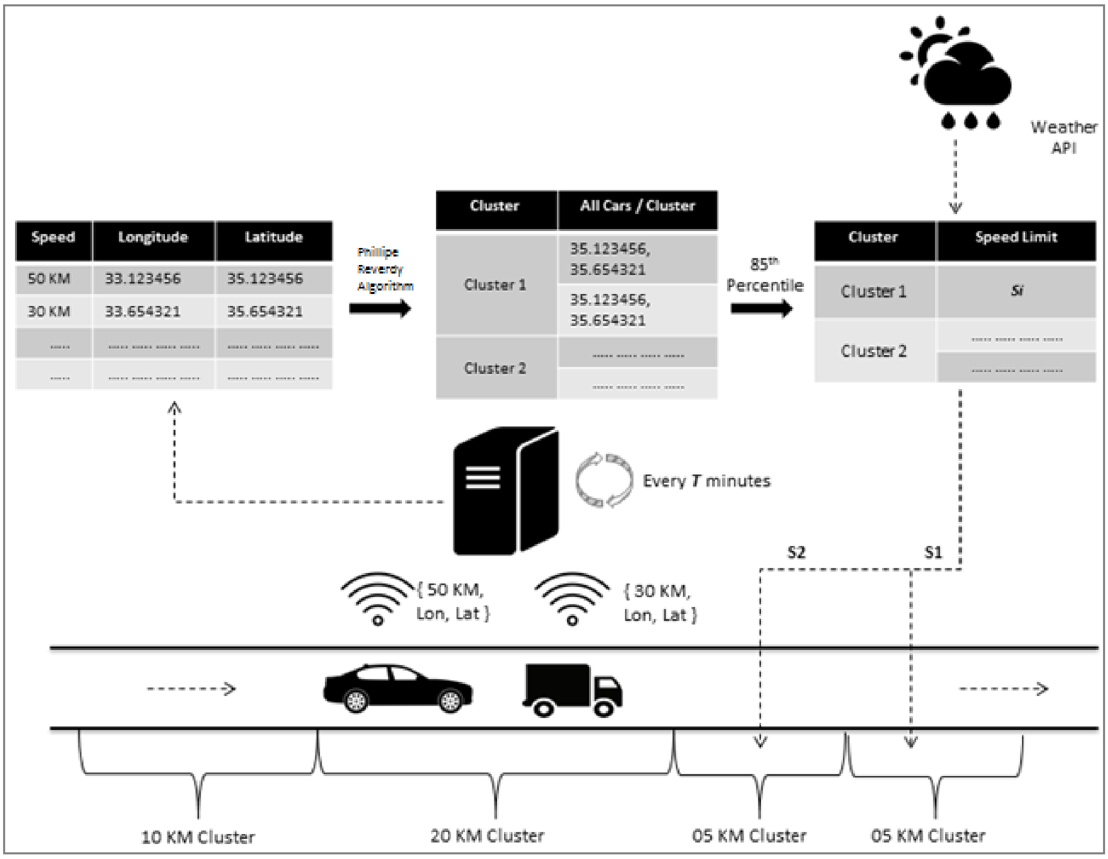

Figure 6. *Adaptable Pattern*

Weather conditions: Another variable used in the adaptable model is the current weather conditions. The vast majority of most weather-related crashes happen on wet pavement and during rainfall: 73% on wet pavement and 46% during rainfall [13]. In this system, we relied on "Highway Capacity Manual 2000" Chapter 22 as the source for changing the speed according to the weather. The following table summarizes the reduction rates:

Weather Conditions	***Freeway Traffic Flow Reductions***			
	Average Speed	**Free-Flow Speed**	**Volume**	**Capacity**
Light Rain/Snow	3% - 13%	2% - 13%	5% - 10%	4% - 11%
Heavy Rain	3% - 16%	6% - 17%	14%	10% - 30%
Heavy Snow	5% - 40%	5% - 64%	30% - 44%	12% - 27%

Table 1. *Speed limit reduction rates according to weather conditions (Source: U.S. Department of Transportation:* Road Weather Management Program)

As a matter of fact, the server contacts the Weather Underground API to acquire the weather forecast of the current cluster and consequently, it reduces the speeds accordingly.

5 Results

In this section we focus on showing the results of only the adaptable model since the static model relies on fixed speed limits and does not depend on dynamic speed detection. To validate the results, we developed an Android mobile application to determine the vehicles' speed within a selected cluster of 06 Kilometres (Figure 7) between two Lebanese towns namely, Jounieh and Jbeil. This mobile application was distributed to 150 drivers using a link on a file server. We used the mobile application knowing that it is infeasible to manufacture 150 IoT devices considering this method is still under research.

Figure 7. Jounieh – Jbeil Highway Cluster (Lebanon)

The following table exhibits the 150 average speeds calculated by the server. The mobile app publishes the speed to the server every one second; consequently, the server calculates the average speed per car every ***T*** seconds (preconfigured per country).

Speed in Km/h collected from 150 vehicles														
55	85	75	65	95	50	75	40	70	55	73	85	85	85	70
40	40	95	95	85	60	100	70	100	40	100	90	100	100	35
80	65	85	55	40	35	85	75	80	35	80	75	100	95	35
80	90	90	50	80	35	40	50	55	40	40	35	95	35	65
35	72	70	75	65	35	40	45	50	90	40	95	35	55	65
85	80	80	95	85	35	60	80	90	80	70	70	65	60	55
90	40	40	50	95	80	80	35	35	65	60	85	90	45	85
65	75	95	65	60	85	90	45	45	60	50	55	35	65	60
90	90	35	100	70	60	80	35	80	35	77	45	85	95	95
40	70	55	35	50	75	40	60	35	80	60	95	55	80	100

Table 2. Speed collected from the mobile applications distributed to 150 drivers.

For better graph visualization, we calculated the frequencies of the speeds by dividing them into class intervals of width of 05 Km/h each as shown below.

35-39	**40-44**	**45-49**	**50-54**	**55-59**	**60-64**	**65-69**	**70-74**	**75-79**	**80-84**	**85-89**	**90-94**	**95-100**
19	14	5	7	9	10	10	5	8	15	13	10	20

Table 3. Frequency table of speeds per class.

Applying the 85th percentile theorem stated in section 4.3.2 on the above frequency table, the resulting speed limit showed 90.0 Km/h.

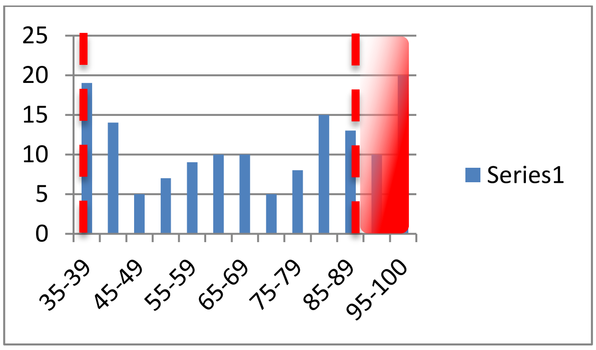

Figure 8. Jounieh – Jbeil Highway Cluster (Lebanon)

The server correlates the maximum speed limits to the current corresponding weather condition, applies the speed reduction rate as depicted in Table 1, and deduces the final maximum speed limit.

As a result, our IoT device considers 90 Km/h as the speed limit and an embedded customized panel displays the maximum allowed speed limit.

6 Performance

This IoT device could have been implemented using the HTTP protocol instead of MQTT. But since HTTP is a stateless protocol in which each client request is treated as an independent transaction that is unrelated to any previous request, many additional technical requirements are then needed to maintain session awareness between the client and the server; therefore, more system resources and networks capabilities are required. That would lead to poor system performance, memory leaks, and power drain.

Unlike HTTP, the adopted MQTT is a bidirectional IoT protocol which maintains stateful session awareness. What differentiates MQTT is its lightweight overhead; it requires minimal bandwidth and less power consumption. Thus it helps minimize the resource requirements for the IoT device and ensures reliability and some degree of assurance of delivery with grades of service.

The table below shows a comparative test of a certain number of messages sent and received over HTTPS (HTTP secure) or MQTT and the battery power consumption per message in 3G and WiFi has been recorded.

This test had been conducted by using a smartphone running Android 2.2, and was simulated by sending 1024 messages, of one byte each, to and from the mobile device. Results showed tremendous advantages of the MQTT protocol over the HTTP [14].

		3G		Wifi	
		HTTPS	**MQTT**	**HTTPS**	**MQTT**
Receive	Msgs / hour	1,708	160,278	3,628	263,314
	% battery / msg	0.01709	0.00010	0.00095	0.00002
	Msgs (note losses)	240 / 1024	1024 / 1024	524 / 1024	1024 / 1024
Send	Msgs / hour	1,926	21,685	5,229	23,184
	% battery / msg	0.00975	0.00082	0.00104	0.00016

Table 4. The following table compares messages per hour, and battery usage per message, between HTTP and MQTT networks.

7 Conclusions and Future Work

In this paper, we proposed and implemented an innovative IoT system which may help the community reducing the death rates resulting from high speed vehicles crashes. It is worth mentioning that the use of this IoT system is restricted for governmental use only due to some privacy concerns related to the drivers' identity and vehicles' tracking. Two models were successfully implemented to detect the speed limit and verified in real environment.

This autonomous system also assists governments in issuing car tickets, collecting fines, controlling the speed limit, and reporting road conditions without human interaction. The results showed that this system is robust and efficient due to the adoption of the MQTT lightweight IoT protocol. Results also showed that system helps the government in: (a) Practicing full control on traffic monitoring and enforce speed laws in all roads and not only highways. (b) Issuing real time tickets without any human interaction. (c) Dismissing traditional devices like RADAR, LIDAR, and SDCS, and thus saving remarkable amount of money. (d) Generating daily records related to the traffic state, to the number of issued tickets in all over the country or in a specific area. These records are stored instantaneously into a dedicated database and can be later fed into a data mining engine for future statistical analysis. (e) Producing daily monetary reports related to fine amount. (f) Customizing the overspeed threshold value conferring to each country's speeding policy and procedures. (g) Reducing the number of policemen in the road. (h) Decreasing the maintenance cost compared to traditional systems. (i) Adopting inexpensive solution to traffic control and traffic issuing. (j) Reducing the number of accidents and lessening the mortality and morbidity rate.

As future work, our intention is to update/upgrade/enhance this system to become an aggregated unit and to embed it into every vehicle during the manufacturing phase. Also, future dynamic road clustering allocation problem is one of our key factors to solve.

References

[1] Who.int « Road safety - Speed ». [online] Available at: http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/speed_en.pdf

[2] ASIRT. « Annual Global Road Crash Statistics », [online] Available at: https://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics

[3] Bole & Wall & Norris (2013) *« Radar and ARPA Manual, 3rd Edition ».*

[4] Weitkamp (2005) « Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere »

[5] Laser Jammers (2012). *How Laser Jammers Work*. [online] Available at: http://www.laserjammer.net/2012/how-laser-jammers-work/

[6] Bhatkar, Shivalkar, Tandale, Joshi. *« Survey of Various Methods used for Speed Calculation of a Vehicle »*

[7] Peter Friess (2013) *« Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems ».*

[8] Ericsson (2016) *« Cellular networks for massive IoT »* [online] Available at: http://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf

[9] Alfred Leick (1995) *« GPS Satellite Surveying 2nd »*

[10] Hasan Al-Jabbouli (2011) *« Data clustering using the Bees Algorithm and the Kd-Tree structure»*

[11] Texas Department *« Procedures for Establishing Speed Zones Manual »* [online] Available at: http://onlinemanuals.txdot.gov/txdotmanuals/szn/szn.pdf

[12] Paul Bourke (1987) *« Determining if a point lies on the interior of a polygon »* [online] Available at: http://masters.donntu.org/2009/fvti/hodus/library/article2/article2.html

[13] U.S. Department of Transportation *« Road Weather Management Program »*

[14] IBM. MQTT: Enabling the Internet of Things. [online] Available at: https://www.ibm.com/developerworks/community/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/tc_overview?lang=en