Dynamic Vehicular Routing with Pollution Metric using Internet Of Things

POOJAH G, KANISHKHA L D, RAJ VIGNESH K, EDNA ELIZABETH N, KAYTHRY P
Department of Electronics and Communication Engineering
SSN College of Engineering
Old Mahabalipuram Road,Kalavakkam, Tamil Nadu 603110
INDIA

Abstract **:** Vehicular exhausts are the root cause of air pollution in every major city around the world. According to the Environmental Protection Agency (EPA), motor vehicles contribute to 75 percent of Carbon Monoxide (CO) pollution in the United States. Pollutants from automobiles, like CO, compounds of Oxides of Nitrogen ($NO_{x)}$ etc., are all major contributors to the overall air pollution of the environment. Air pollution monitoring is practiced by power plants, which is a very expensive activity. The objective of this work is to navigate the least polluted routes for automobiles to reach the desired destination at a minimal cost.To achieve the goal and to reduce the cumulative pollution caused by automobiles, sensors are used to compute the magnitude of the air pollution in an automobile-intensive area. The pollution data collected by the sensors is sent to a cloud-based platform known as Firebase. Subsequently, a mobile application created using Android Studio is used to integrate the Firebase data and Google Maps.The novelty of this paper is to find the route dynamically which has the least pollution level, using the developed android application.

Key-Words:Arduino, Firebase, Google Maps, Android Studio, IoT

Received: June 11, 2021. Revised: April 24, 2022. Accepted: July 12, 2022. Published: September 26, 2022.

1. Introduction

Climate change is real and the exorbitant rise of greenhouse gases and pollutants are testament to that. After a certain threshold, these have the potential to create a huge negative change in our environment. Hence, an answer for this issue will help in steering humanity towards the path of safe and clean surroundings. The major source of air pollution is contributed by the automobile industry in the urban areas of developing countries. Especially in countries like India, low initial cost motorized vehicles are commonly used by citizens. The number of vehicles is increasing by the day with limited emission controlling technologies which is the prime reason behind air pollution [1].The major pollutants emitted through vehicular sources are Carbon Monoxide (CO), Hydrocarbons (HC), Oxides of Nitrogen (NO_x), Particulate Matter (PM), lead (Pb) and Sulphur dioxide (SO_2). The air quality in the megacities throughout the world is exceeding the prescribed limit. Power plants were used to monitor the pollution level, which is an expensive method. The objective of the project is to determine and maintain the least possible pollution, along the route emitted by vehicular exhausts.

By the usage of gas sensors which can detect CO, NO_x etc. The pollution levels are monitored in certain areas where it is present in huge magnitude. These

sensors are used to send data to Firebase, a cloud computing platform for database management systems. This research is in the realm of Internet of Things (IoT), where the real time dynamic data is sent to Firebase via Internet communication. IoT refers to uniquely identifiable objects and their virtual representation in an Internet-like structure. It gives the research on monitoring methods, a wider platform and much more possibilities.

Some of the latest researches in the field of Pollution monitoring using gas sensors have not yet resulted in a direct application to the general public. K.K. Khedo, R. Perseedoss, and A. Mungur developed an air pollution monitoring system using wireless sensor network. The system was deployed at selected location in Mauritius to monitor the air pollution level around the city [2]. But the authors do not parameterize the sensor data for further usage.

T Bektas and G Laporte proposed the Pollution-Routing Problem (PRP). In PRP, vehicular emissions are quantified to be a metric along with distance, speed and load factor for vehicular routing [3]. PRP is significantly complex to solve optimally. Also, the proposed PRPmethodology does not take the pollution existing in the environment into account, for finding the route.

S.Kaivonen and E.C.H. Ngai proposed the idea of deploying air pollution sensors on city buses to read the pollution data more accurately as compared to a stationary sensor [4].With the sensors, they tried to monitor and collect the air pollution data throughout the country of Sweden. Similarly, CO2 is monitored in the urban area [5]. But the authors did not consider the problem of vehicular routing.

From the existing literature we found that there is room for integrating the pollution level as the routing metric. Ambient air pollution level data sensed by the sensor is provided onto the Google maps using a middleman application, Firebase. The commuters can select the least polluted route using the Google maps. The users can also access the pollution data from any of the sensor nodes deployed in the area which is highlighted in the map. In this paper we presented a system, equipped with sensors, cloud platform and Google Maps to find the least polluted route to reach a destination among all the available paths.

The remainder of this paper is organized as follows. The details of system methodology are given in Section 2. The hardware and software part of the system is implemented in section 3. The proposed pollution level-based routing algorithm is presented in section 4. The real time implementation results in terms of less polluted routing are discussed in section 5. Finally, we conclude the paper in section 6.

2. Methodology

The procedure for accomplishing the work proposed in this paper has diverse phases. Those phases comprise of collection of sensor data, transferring it to Firebase, accessing the data through an Android application and using it as parameters for routing purposes, as shown in Fig 1.1.

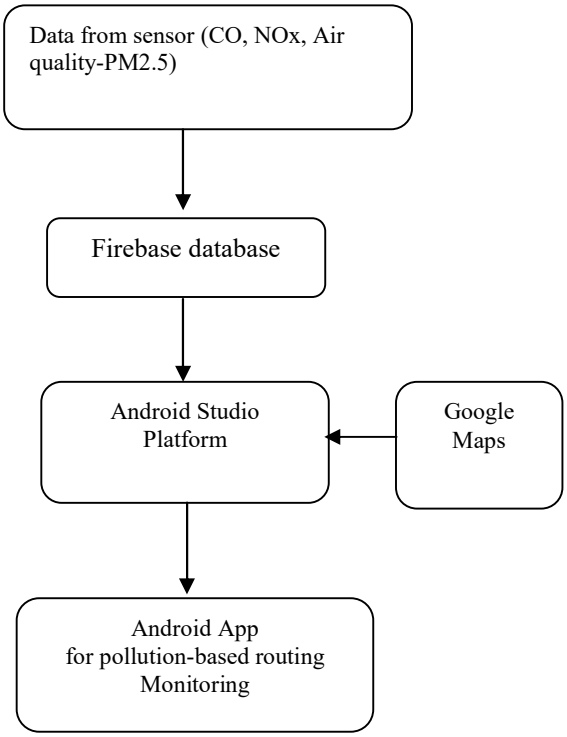


Fig 1.1 Least polluted route identification process Flow

The components that are considered to affect the atmosphere are, as tabulated below, in Table 1.1. Gravimetric analysis can be used to find the amount of particulate matter in air. It has been incorporated as per the National Ambient Air Quality Standards NAAQS) manual published by Central Pollution Control Board (CPCB) in 2011.

Table 1.1 Equations from Gravimetric analysis

Component	Equation
PM2.5	$M_{2.5} = (M_f - M_i)\ mg \times 10^3\ \mu g/V$
PM10	$M_{10} = (M_f - M_i)\ mg \times 10^3\ \mu g/V$
NOx	$(A_s - A_b) \times C_f \times (V_s/V_a) \times V_t \times 0.82\ mg/ml$
SOx	$(V_1 - V_2) \times N \times K/V\ mg/ml$

Where,

$M_{2.5}$ = Total mass of fine particulate collected during sampling period (µg)

M_f = Final mass of the conditioned filter after sample collection (mg)

M_i = Initial mass of the conditioned filter before sample collection (mg)

10^3 = Unit conversion factor for milligrams to micrograms

V = total sample value (m^3) $(Q_{avg} \times t \times 10^{-3}\ m^3)$

Conventionally, these parameters are recorded by large power plants, which run on coal [6]. Some disadvantages of those power plants are: large amount of space is occupied, the time taken to construct the plants is huge and is very expensive to build and set up. The methodology proposed in this paper solves these issues by replacing huge power plants with small air quality sensors. The deployed air quality sensors yield similar results as compared to power plants. It is very cost effective and environment friendly.

PM2.5 is considered to be the standard of air quality measurement, according to EPA. Hence, we adapt the same in our work as well. The air quality sensor, GP2Y1014AU sensor is used to sense the PM concentrations in the atmosphere.

The air quality data sensed by the sensor nodes are then sent to the cloud. These air quality data are stored in the database created using Firebase, a cloud platform by interfacing the sensors with a ESP8266 Node Micro Controller Unit (NodeMCU) module. ESP8266 NodeMCU module is preferred as it has a built-in Wi-Fi Module which makes it less expensive and simple to use as compared to Raspberry Pi module [7-9]. Firebase cloud platform is preferred because of its compatible features with Android studio, which is used to create an application.

The application's functionality is designed such that it requires the user to mention the point of departure and the point of arrival. The app identifies the least polluted path available between the origin of the user and destination. In the map, the user can view the least polluted route is highlighted among all the paths available between origin and destination of the user.

3. Implementation

The hardware and software implementation of the proposed method is discussed in detail in this section. In the hardware implementation part, the transmitting module is constructed on a Printed Circuit Board (PCB). The ESP8266 module, is connected to a personal computer and programmed to send the real-time sensor data to the cloud-based platform, Firebase. With respect to the software implementation the stored data is read from Firebase by an android application developed using android studio, which is connected to google maps. The android application employs a routing algorithm to provide the user with the least polluted path based on their source and destination on Google Maps.

3.1 Hardware implementation

The nodes are deployed in selected location of the chosen area. The node deployment location points are

decided based on their proximity to each other and by its approximate pollution level. The connectivity between the deployed nodes in the chosen area can be visualized together in the form of a connected graph.

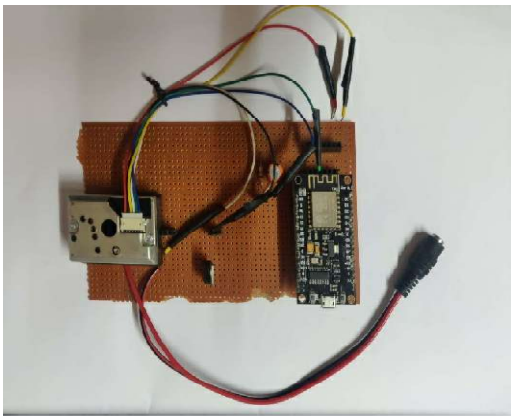


Fig 1.2 Hardware implementation of transmitting module

As mentioned earlier, each transmitting module consists of a GP2Y1014AU sensor, an ESP8266 Wi-Fi module, a voltage regulator, capacitors and a DC supply as shown in Fig 1.2. Many sensor nodes can be deployed based on requirement and coverage area. These nodes monitor the air pollution level and aids in finding the least polluted route in the given area. The number of sensor nodes can be increased for more accuracy. For our study we chose our University campus area for the sensor node deployment and real time implementation.

3.2 Firebase

Each sensor node in the transmitting module logs the pollution data in real time scenario on the Firebase server. Firebase is used to manage and modify data generated from any sensors, android application, web services, etc.

Firstly, the credentials of Firebase database and Wi-Fi are defined in the Integrated Development Environment (IDE) of Arduino [10]. This setup is used to connect the Wi-Fi to the NodeMCU. Among the available baud rates in the Arduino, 115,200 symbols per second is chosen to facilitate the fastest serial communication for the module. Then the sensor nodes are connected to the Internet using Wi-Fi, thereby, the pollution values sensed by the nodes are uploaded onto Firebase.

The pollution data, in the form of dust density, is calculated by the sensor from the node's environment. The pollution values are normalized by converting the measured signal voltage into a scaled voltage value and then to its corresponding dust density value. This process occurs continually to sync real time data with Firebase. Because the nature of the environment keeps changing and consequently, the air quality is constantly varying.

3.3 App – initiation, linking with Firebase and Google Maps

The pollution data from the sensor is stored in Firebase. Information from all the nodes is examined to compute the least polluted path. To realize this purpose, the sensor data must be accessed by the developed android application. To proceed further, a *json* file is created by Firebase and is placed in the directory of the application project inside the Android Studio. The *json* file is essential for the Android Studio to comprehend the files of Google Maps and Firebase.

Once the json file is created, the building tasks of the application are automated using the *build.gradle* script [11]. Gradle is an open-source build system which is used to run the application on the device. The gradle is required to add class paths under the dependencies block. Dependencies are used to link Firebase and Google Maps to the Android Studio. The plugins are used to read the *json*file and will inject the values in the build of the application. Subsequently, the sub codes are automatically generated to facilitate the functioning of the application.

3.4Software Implementation

The linkage between the various components on a software level is accomplished by the usage of *build.gradle* files. By clicking the location marker on

the application, the user can view the pollution level at a specified location. This pollution data is retrieved from the firebase database, with the link established by *json* file [12,13].

The latitude and longitude of the deployed node locations are programmed into the application. Using these positional parameters, a marker is created to point the node location and to display the data. The pollution data retrieved from the firebase is fed into the routing algorithm to find the least polluted path.

4. Proposed Routing Algorithm based on air pollution level

In this section the we discussed the Vehicular routing algorithm by considering air pollution level as one of the metrics. This algorithm is quite impressive because it balances the shortest path as well as the least polluted route finding. The user mentions the point of origin and point of destination in mobile app. The app search for those locations in the Google Maps and instruct the user to take a particular path. A searching algorithm is developed such that it searches for the least polluted path between the source and destination point by referring with air pollution level reported by the respective sensor nodes deployed near the path. Fig 1.3 shows the graphical representation of the nodes implemented for our study.

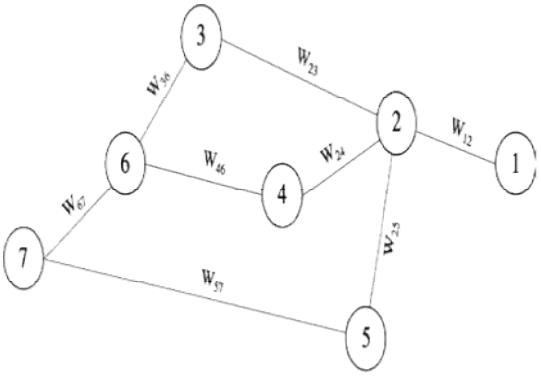


Fig 1.3 Graphical Representation of the nodes

The parameters involved in this algorithm:

1. Pollution value, P_i– Pollution value of i^{th}node.
2. Distance between two nodes, D_{ij}– Distance between node i and j.

Weight estimation between two nodes:

The pollution P_i can be calculated using Equation (1).

$$Calc\_Voltage = Meas\_Voltage \times (5.0/1024) \quad (1)$$

$$P_i = 0.17 \times Calc\_Volt age - 0.1 \quad (2)$$

where, the *Meas_Voltage* is the voltage level measured from the pollution sensor and ranges from 0-5V.

Normalizing the pollution value:

$$P_{norm\_i} = P_i / P_{\max} \quad (3)$$

where, P_{max} is the maximum value that can be obtained from the pollution sensor at *Meas_Voltage* = 5V.

Hence,the routing weight between two nodes, W_{ij}, can be calculated using the equation, Eq. 4.

$$W_{ij} = (P_{norm\_i} + P_{norm\_j})/2 \times D_{ij} \quad (4)$$

There are two steps involved in the routing protocol: Once the user enters the destination point in the app, the first step is to find all the available paths between the user's starting point and the destination point. The routing is done by incorporating, not only pollution data but also the distance between the source and destination, as a metric. The second step is to find the weight between the sensor nodes along those paths. The weight W_{ij} is calculated using the normalized pollution value (P_{norm_ij}) and distance (D_{ij}) between the nodes. The path with least summation of weight is identified as the least polluted route. The distance is made a factor to eliminate extreme cases where a relatively large distance has to be travelled in order to take the least polluted path. This is done to ensure that no compromise is made between pollution stability and fuel wastage. The process of finding a least polluted possible shortest route is shown in Algorithm 1.

Algorithm 1 Least Polluted routing

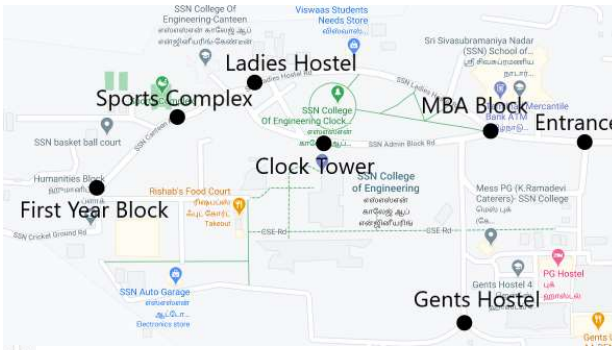
Least Pollution Path (G, start, destination)

Input:A undirected pollution weighted graph G with non-negative edge weights and a start vertex, start and end vertex, destination.
Output: Path from start to destination nodes.
Initialize *D(start) = 0* and *D(i) = Inf* for *i!= start*
Initialize priority queue Q of all vertices in G using D as the key.
while Q is not empty do
u = Q.removeMin()
for each vertex j adjacent to i and in Q do
if D(i) + W((start, j)) < D(j) then
D(j) = D(i) + W((i, j))
update j in Q
append j to Path
return Path, D(destination)

---

5. Results and Discussions

For the purpose of real time implementation, seven sensor node modules were deployed at various locations with the coverage area of 1600m x 900m in our college campus. The sensor nodes are static once deployed in the area. The transmission range is 100 m for all nodes. These nodes are deployed approximately at equal distance with majorly covering the all the paths in the area of implementation. These nodes monitor the air pollution level and aids in finding the least polluted route in the given area. Seven sensor nodes were placed across different points in the college's campus, as shown in Fig 1.4.



- Deployed Sensor node

Fig 1.4 Sensor nodes in the college campus

The path between the starting point and destination is shown based on the distance between the two points and the air pollution along the available paths. The sensors measure the PM2.5 value at their respective locations along the paths, which is shown in Table 1.2, whose sum determines the cumulative pollution value for the path, in accordance with Algorithm1.

Table 1.2 The PM2.5 values of each node

Node	PM2.5 Value (µg/V)
Gate	10
MBA	7
Clock Tower	12
Ladies Hostel	6
Sports Complex	9
First Year Block	4
Gents Hostel	7

Conventional routing algorithm establishes the shortest routing path based on the distance factor, where pollution values are not considered. For our implementation, we have considered the starting point to be "Entrance" and the destination to be "First Year block". The path shown is (Path1) Entrance (source) via MBA Block, Clock Tower, Sports Complex, First Year block (destination) as shown in Fig 1.5.

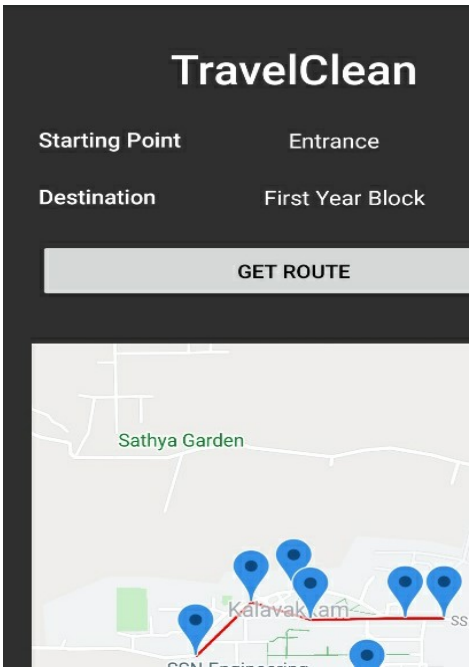


Fig 1.5 Distance based Routing (Path1)

For the same origin and destination points, the proposed routing algorithm considering the pollution data and distance, determines the path (Path2): Entrance (source), MBA Block, Gents Hostel, First Year block (destination), as shown in Fig 1.6.

Though the distance difference between the two paths is negligible, the inclusion of the pollution metric favors the path via Gents hostel since the priority is identifying pollution less path to travel. Table 1.3 indicates the path chosen by the conventional and proposed routing algorithm in terms of pollution and distance parameter respectively.

Table 1.3 Pollution vs Distance for different paths

Path	Distance (km)	Pollution (µg/V)
Path1	1.1	48
Path2	1.85	28

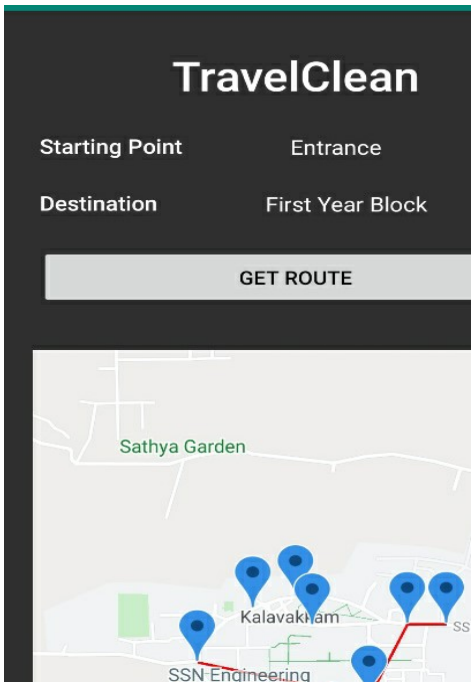


Fig 1.6 Pollution and Distance based Routing (Path2)

6. Conclusion

Air quality sensors were interfaced with an ESP8266 module each, in order to create a pollution monitoring node. These nodes were connected, via the Internet, to the Firebase platform to create a real-time pollution database of every individual node's environment. Subsequently, an Android application was created using Android Studio, which tethered the application and the aforementioned database. The function of this application is to provide the least polluted path for a user, based on the source and destination given, using a dynamic routing algorithm.

Routing is performed by taking the pollution data available in Firebase and the distance between the source and destination as input parameters. Hence, resulting in an algorithm which balances both distance and pollution factors to provide the solution. There is no sizable trade-off between pollution and distance travelled. Hence, the everyday standards of living of the individuals utilizing the application are tremendously improved with respect to their health.

References

[1] Aditya Kumar and SushantaTripathy, "Study of Vehicular Pollution and its Mitigation Measures" *3rd KIIT International Symposium on Advances in Automotive Technologies,* KIIT University, Bhubaneswar, 2014.

[2] K.K. Khedo, R. Perseedoss, and A. Mungur, A Wireless Sensor Network Air Pollution Monitoring System, *International Journal of Wireless and Mobile Networks (IJWMN),* Vol 2, No. 2, May 2010

[3] T. Bektas and G.Laporte, The Pollution-Routing Problem, *Transportation Research Part B: Methodological,* Vol 45, Issue 8, September 2011

[4] S. Kaivonen and E.C.H. Ngai, Real-Time Air Pollution Monitoring With Sensors On City Bus, *Digital Communication and Networks,* Vol 6, Issue 1, February 2020.

[5] X. Mao, Xin Miao, Yuan He, X. Li, Y. Liu, CitySee: Urban CO2 monitoring with sensors, *Proceedings IEEE INFOCOM* , 2012.

[6] https://oizom.com/case-study/gmr-power-plant-pollution-monitoring/

[7] https://circuitdigest.com/microcontroller-projects/iot-firebase-controlled-led-using-esp8266-nodemcu

[8]https://github.com/FirebaseExtended/firebase-arduino/tree/master/examples/FirebaseDemo_ESP8266

[9]https://circuitdigest.com/microcontroller-projects/sending-temperature-and-humidity-data-to-firebase-database-using-esp8266

[10]https://android.jlelse.eu/android-firebase-authentication-with-google-signin-3f878d9b7553

[11]https://console.firebase.google.com/u/0/project/fir-demo-757dc/overview?pli=1

[12]https://stackoverflow.com/questions/39394502/databasereference-and-firebasedatabase-cannot-resolve-symbol

[13]https://github.com/pjwelcome/GoogleMapsDirections/blob/master/app/src/main/java/com/multimeleon/android/googlemapsdirections/MapsActivity.java

[14]https://www.sparkfun.com/datasheets/Sensors/gp2y1010au_e.pdf