



ADVANCE POLLUTION MONITORING SYSTEM WITH HEALTH CRITICAL GAS INDEXING USING IOT

Mr. Ashish Manusmare, Ms. Pooja Jayant Gajarlawar
Department of Electronics & Communication Engineering
Ballarpur Institute of Technology, Ballarpur

Abstract— The rise of urbanization and industrialization has led to an increase in pollution, posing serious risks to human health and the environment. Consequently, the development of advanced pollution monitoring systems has become crucial for efficient pollution control and management. This abstract introduces a cutting-edge pollution monitoring system based on the Internet of Things (IoT), which incorporates health critical gas indexing to provide real-time and precise information about air quality. The proposed system utilizes IoT technology to create a network of interconnected sensors installed in various locations within cities or industrial areas. These sensors are designed to measure the concentration of pollutants such as carbon monoxide (CO), nitrogen dioxide (NO₂), sulfur dioxide (SO₂), particulate matter (PM_{2.5} and PM₁₀), and ozone (O₃) in the surrounding air. The collected data is wirelessly transmitted to a central monitoring station for analysis and visualization. To assess the health risks associated with measured pollutant levels, a health critical gas indexing algorithm is employed. This algorithm considers pollutant concentrations and their corresponding health effects based on established air quality guidelines and standards, calculating an index value that indicates the overall health impact on the exposed population. The monitoring system provides real-time updates and alerts to relevant stakeholders, including government authorities, environmental agencies, and the public. Accessible through a user-friendly web or mobile application, the data and index values empower users to make informed decisions regarding outdoor activities, pollution mitigation measures, and health precautions. Compared to traditional monitoring approaches, the IoT-based advance pollution monitoring system with health critical gas indexing offers several advantages. It provides a comprehensive and accurate assessment of air quality by considering the specific health risks posed by different pollutants. Its real-time nature enables prompt actions and interventions to mitigate pollution levels, safeguarding public health and the environment. Furthermore, the availability of data empowers

polymakers and researchers to analyze trends, identify pollution sources, and develop targeted pollution control strategies. In conclusion, the proposed IoT-based advance pollution monitoring system with health critical gas indexing represents a significant advancement in pollution monitoring and management. By leveraging IoT technology and incorporating health-related considerations, the system enables effective and proactive pollution control measures, leading to improved air quality and overall well-being of the population.

Keywords—Drowsiness Detection, Image Processing, Matlab, Eye Detection

I. INTRODUCTION

In recent times, there has been a significant surge in the number of vehicles worldwide, resulting in a substantial increase in air pollution caused by vehicular emissions. This has become a pressing concern for both the environment and public health, necessitating the development and implementation of effective solutions to counter the detrimental effects of such pollution. One solution that has emerged is the Vehicular Pollution Management System, which is a comprehensive framework designed to monitor, control, and minimize pollution stemming from automobiles[1]. This advanced system integrates various components and cutting-edge technologies, employing data analytics and policy interventions to address the complex challenges associated with vehicular pollution.[2] The primary objective of the Vehicular Pollution Management System is to regulate and monitor vehicle emissions, ensuring compliance with emission standards set by regulatory bodies. It accomplishes this through a network of monitoring devices, strategically placed in urban areas, highways, and relevant locations, including emission testing equipment, sensors, and cameras. These devices capture real-time data on vehicle emissions, including pollutants like carbon monoxide (CO), nitrogen oxides (NO_x), particulate matter (PM), and hydrocarbons (HC) [3]. The collected data is transmitted to a centralized control center, where it undergoes processing and analysis using

advanced algorithms and machine learning techniques. This enables the system to generate accurate and up-to-date information on the emission levels of individual vehicles, specific routes, and areas with high pollution levels.[4] The analysis of this data provides valuable insights into the sources and patterns of vehicular pollution, facilitating evidence-based decision-making and targeted interventions.[5]

To enforce emission standards and discourage non-compliant vehicles, the Vehicular Pollution Management System includes an automated enforcement mechanism. It integrates with vehicle registration databases and the relevant authorities responsible for issuing licenses and permits. This integration allows for real-time identification of vehicles that fail to meet emission standards or lack necessary certifications. Violations are swiftly detected, leading to automatic imposition of penalties or fines, thereby encouraging vehicle owners to adhere to pollution regulations. Additionally, the system incorporates public awareness and education initiatives to foster a sense of environmental responsibility among vehicle owners and users.[6] It utilizes various communication channels, such as mobile applications, websites, and public campaigns, to disseminate information about the adverse effects of vehicular pollution and the significance of emission control measures [7]. These efforts aim to empower individuals to make informed choices and adopt sustainable transportation practices.

Furthermore, the Vehicular Pollution Management System supports policy interventions and measures to tackle pollution at its root.[8] It provides decision-makers with comprehensive data on the effectiveness of existing policies and strategies, enabling them to assess their impact and make informed adjustments.[9] This data-driven approach facilitates the development and implementation of innovative measures, including the promotion of electric and hybrid vehicles, improvement of public transportation infrastructure, and incentivization of eco-friendly practices. Overall, the Vehicular Pollution Management System represents a crucial step towards achieving sustainable urban mobility and mitigating the adverse effects of vehicular pollution. By leveraging technology, data analytics, and policy interventions, this system offers a holistic approach to monitor, control, and reduce pollution caused by automobiles.[10] Its potential to transform cities into cleaner, healthier, and more livable spaces holds the promise of a sustainable future for generations to come.[11]

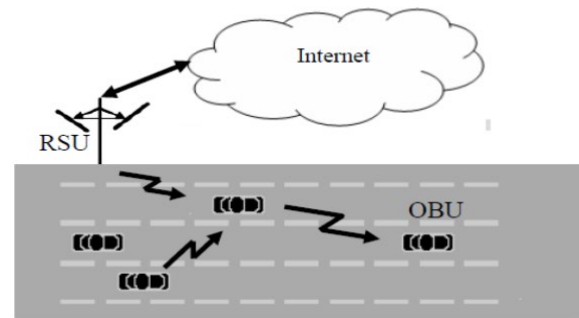


Figure.1 Node types in VPMSs

- **Health Critical Gas Indexing**
Health Critical Gas Indexing is a methodology utilized to evaluate the potential health risks of various gases by considering their toxicity, concentration levels, and exposure duration. The purpose is to prioritize and classify gases based on their capacity to harm human health. By assigning an index to each gas, it becomes easier to compare and prioritize them, aiding organizations and regulatory bodies in allocating resources and implementing appropriate safety measures. The process involves evaluating the toxicity level of each gas, considering the concentration levels, and factoring in the duration of exposure. Gases with higher toxicity or concentration levels, as well as those posing immediate harm even with short-term exposure, are assigned higher index values. Once ranked, the index values can be presented in different formats to facilitate understanding and decision-making.[12] Health Critical Gas Indexing provides a systematic approach to manage risks, enabling prompt identification and addressing of critical health hazards. It plays a vital role in developing safety protocols, setting exposure limits, and implementing preventive measures to safeguard human health in diverse settings.[13]

II. METHODOLOGY

- **Modeling & Protocol Generation**
This project focuses on mitigating upstream pollution in a Wireless Sensor Network (WSN) that utilizes single-path routing. Figure 3-1 illustrates the architecture, where sensor nodes within the network continuously generate particles and transmit data in a many-to-one convergent traffic pattern in the upstream direction. These sensor nodes are assumed to implement a protocol similar to the Internet of Things (IoT) Virtual Power Management System (VPMS). Each sensor node in the network can generate two types of traffic: source and transit. Source traffic is locally generated at each sensor node, while transit traffic originates from other nodes within the network. Consequently, each sensor node can function as a source node, an intermediate node, or both. If a sensor node has both offspring nodes and transit

traffic, it serves as both a source node and an intermediate node. Conversely, if a sensor node does not have any offspring nodes and only generates source traffic, it is solely a source node. The offspring node of a specific node is defined as the node whose traffic is routed through that particular parent node. If an offspring node establishes a direct connection with its parent node, it is referred to as a child node, while the parent node is known as the parent of the child node.

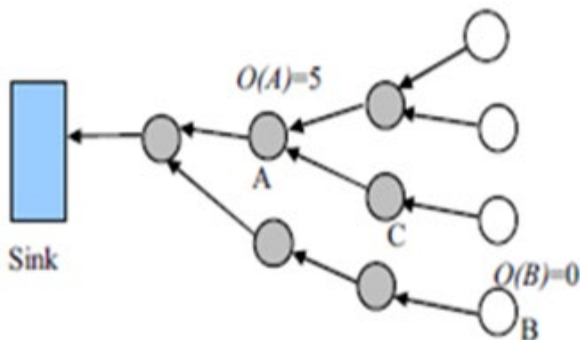


Figure.2 Environment model-logical topology established by routing protocol

Figure 2 depicts a network comprising nodes, with a particular focus on node A. It is noteworthy that node A exhibits a distinct characteristic by having five offspring nodes linked to it. Consequently, node A assumes a dual role within the network, serving as both a source node, responsible for generating data or information, and an intermediate node, responsible for transmitting this data to other nodes.[14] Thus, node A plays a crucial and versatile function in the network. Additionally, node C is identified as a child node of node A, indicating that node A serves as the parent node for node C. In simpler terms, node C directly receives information or data from node A, establishing a direct connection within the network hierarchy. Conversely, node B lacks any offspring nodes, and thus solely operates as a source node. It generates data or information without transmitting it to any other nodes in the network. To refer to a specific sensor node, denoted as node i , the notation $O(i)$ is utilized to express the total number of offspring nodes associated with it. It is important to note that sensor nodes typically fulfill the role of both a source node and an intermediate node simultaneously, unless stated otherwise.

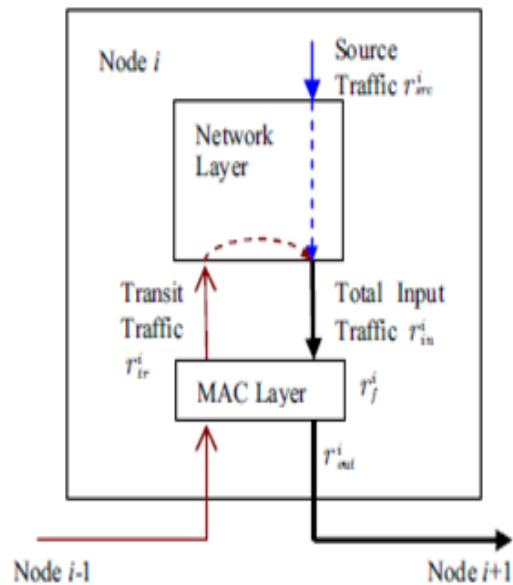


Figure 3. General node model

The paragraph describes a queuing model implemented at a specific sensor node labeled as " i " using single-path routing. The node receives transit traffic from its child nodes and local source traffic, which converge at the environment layer before being forwarded to the next node in line. The model addresses the issue of pollutants accumulating in the Virtual Path Management System (VPMS) layer when the total input traffic rate exceeds the pollutants forwarding rate. The forwarding rate depends on the VPMS protocol and can be influenced by factors like the number of active sensor nodes and their traffic density. If the input pollutants rate is smaller than the forwarding rate, the outgoing pollutants rate will be equal to the input rate. However, if the input rate exceeds the forwarding capacity, the outgoing rate will be close to the maximum forwarding capacity. The passage introduces a Pollution Control approach that focuses on adjusting the input rate of pollutants and the forwarding rate to mitigate pollution levels in the network. It suggests reducing the source rate by adjusting the sampling or reporting frequency and indirectly reducing transit traffic by adjusting the rate at the subsequent node. Additionally, it highlights the importance of reducing channel access to prevent link-level pollution. The approach incorporates a scheduler and utilizes a Weighted Queuing algorithm to ensure fairness between source and transit traffic. By adjusting the scheduling rate, the Pollution Control approach achieves efficient pollution control without changing the VPMS protocol parameters, making it compatible with any IoT protocol.

• **Generating the PCCP**

The Pollution Control and Congestion Control Protocol (PCCP) is a specialized solution developed to address the unique challenges and requirements of Wireless Sensor Networks (WSNs). These networks consist of sensor nodes with different priorities and responsibilities, and PCCP ensures weighted fairness by enabling the sink node to receive varying throughputs from sensor nodes in a proportional and equitable manner. The primary objectives of PCCP are to enhance energy efficiency and support Quality of Service (QoS) metrics related to pollutants delivery latency, throughput, and loss ratio. These protocols optimize the network's performance in handling pollutants generated within the WSN by regulating the transmission of pollutant data and allocating network resources accordingly. In WSNs, sensor nodes collect data about pollutants and transmit it to a central sink node. Efficient management of pollutants is crucial due to limited resources such as energy and bandwidth. Pollution Control protocols address this challenge by regulating data flow, prioritizing pollutants, and allocating resources based on the sensor nodes' requirements. Weighted fairness ensures that nodes with different priorities or responsibilities receive a fair share of resources, considering factors like critical roles or strategic placements. Energy efficiency is a key focus, aiming to minimize unnecessary energy consumption and extend the network's lifetime. QoS metrics, including pollutants delivery latency, throughput, and loss ratio, are also considered to ensure effective data transmission.[15]PCCP incorporates mechanisms for achieving weighted fairness in WSNs. It allocates network resources based on the priorities or critical functions of sensor nodes, resulting in a proportional distribution of throughput. Energy efficiency is improved through techniques such as dynamic duty cycling, adaptive transmission power control, and data aggregation, which conserve power and extend the network's lifetime. PCCP optimizes routing and data dissemination algorithms to minimize data delivery latency and manages congestion and resource allocation effectively, enhancing overall network throughput. Reliable data delivery mechanisms, error correction techniques, and congestion control strategies minimize pollutants-related data loss. By addressing the challenges of weighted fairness, energy efficiency, and traditional QoS requirements, PCCP provides an effective solution for pollution control in WSNs, ensuring fair data distribution, prolonged network lifetime, and improved performance.[16]

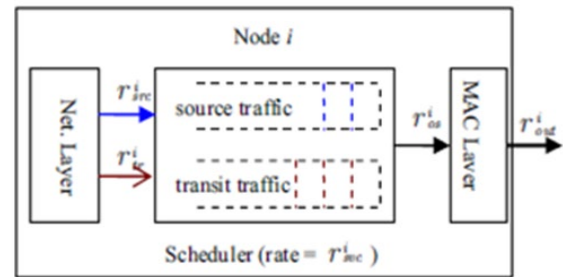


Figure.4 Node model in PCCP

• **Intelligent Pollution detection (IPD)**

In conventional transport protocols such as TCP, pollution detection typically occurs at the endpoints through the analysis of duplicated ACK messages, timers, or the ECN bit in the pollutant's header. However, in sensor environments, intermediate nodes also contribute to pollution detection by monitoring factors like queue length, buffer increment, wireless channel status, or a combination thereof. This implies that a single bit can be employed to convey pollution information. To tackle pollution, one approach involves calculating the sustainable service rate based on the local pollutant's service time and using it to regulate node transmission rates. Nevertheless, these techniques lack precision in accurately reflecting the pollution level, whether at the node or link level.[17] To address this limitation and obtain more accurate measurements of local pollution levels at each intermediate node, we propose an intelligent Pollution Detection (IPD) system. IPD detects pollution by analyzing the mean inter-arrival time (iat) and mean service times (ist) of pollutants at the Virtual Pollution Monitoring System (VPMS) layer. The inter-arrival time refers to the time interval between sequential arrivals of pollutants, either from the source or during transit. On the other hand, the service time represents the time taken for a pollutant to be successfully transmitted from the VPMS layer, including waiting, collision resolution, and transmission times. Both iat and ist can be easily measured at each node for each pollutant individually. Based on these measurements, IPD introduces a new pollution index known as the Pollution Degree (d(i)), which indicates the ratio of the average pollutants' service time to the average pollutants' inter-arrival time over a predefined time interval at each sensor node i:

$$d(i) = t_{si} / t_{ai} \dots\dots (1)$$

The Pollution Degree aims to reflect the current pollution level at each sensor node. When the inter-arrival time is smaller than the service time, the Pollution Degree (d(i)) will exceed 1, indicating pollution at the node. Conversely, if the Pollution Degree (d(i)) is less than 1, it means that the incoming rate is lower than the outgoing rate, resulting in a



decrease in pollution. Therefore, the Pollution Degree effectively represents the pollution condition and provides valuable information for efficient pollution control. Child nodes can adjust their transmission rates using the Pollution Degree ($d(i)$) to determine whether to increase or decrease traffic levels. Equation (1) employs a Weighted Moving Average (WMA) algorithm to measure i_{at} and i_{st} at each node i . When calculating the Pollution Degree, i_{at} is periodically updated whenever N_p (in this case, 3 in PCCP) new pollutants arrive within a time interval, T_{Np} . Here, $0 < w_a < 1$ is a constant (set to 0.1 in PCCP), and within the time interval T_{Np} , N_p new pollutants have arrived. Additionally, t is updated each time a pollutant is forwarded. Similarly, $0 < w_s < 1$ is a constant (also set to 0.1), and i_{st} represents the service time of the just-transmitted pollutant. Regarding transmitting pollution information, there are two main approaches: Explicit Pollution Notification (ECN) and Implicit Pollution Notification (IPN). Each method has its own characteristics and implications. ECN involves using dedicated control messages to convey pollution information, which introduces additional network overhead due to the transmission of these special messages, requiring extra resources and consuming more bandwidth. Conversely, IPN utilizes the header of regular data packets to carry pollution information, eliminating the need for separate control messages. By piggybacking the pollution data in the header, IPN reduces overhead and takes advantage of the broadcast nature of the wireless channel, where child nodes listen to their parent node for pollution information. In the context of PCCP (Preserving Clean Channels Protocol), ICN (Implicit Communication Notification) is employed at each sensor node (i) to incorporate pollution information into the header of particles pollutants for forwarding purposes. Pollution notification is triggered by either of two events: when the number of pollutants forwarded by a node exceeds a predefined threshold, or when the node overhears a pollution notification from its parent node. Once a pollution notification is triggered, the node computes its global priority index, which combines the source traffic priority index of the node itself and the global priority indices of all its child nodes. The source traffic priority index represents the priority level of the node's own data traffic, while the global priority index considers the priorities of both the node and its children. By employing this approach, PCCP aims to efficiently propagate pollution information throughout the network without relying on explicit control messages, thereby reducing overhead and effectively utilizing the broadcast nature of the wireless channel.

• Priority-based Rate Adjustment (PRA)

In Figure 3-3, a scheduler implementation is depicted, consisting of two sub-queues positioned between the environment layer and the VPMS (Virtual Plant Modeling System) layer. By maintaining a scheduling rate, known as r_{sivc} , lower than the VPMS forwarding rate (irf), the output

rate will closely match the output rate $irout$. Thus, adjusting the scheduling rate r_{sivc} enables us to prevent or alleviate pollution. There are two main approaches to achieve this goal. Initially, pollution information is conveyed through a binary Pollution Notification (CN) bit, indicating the presence or absence of pollution. However, precise rate adjustment becomes feasible when nodes receive specific instructions on how much to increase or decrease their rates. In this context, the pollution degree $d(i)$, along with the introduced priority indices $TP(i)$ and $GP(i)$, provide more comprehensive information than the CN bit alone and allow for accurate rate adjustment. To manage pollution at both link and node levels, the Priority-based Rate Adjustment (PRA) technique involves adjusting the scheduling rate r_{sivc} and the source rate r_{sirc} at each sensor node after receiving pollution notifications from their parent nodes. The subsequent section demonstrates a hardware-software interactive modeling approach that effectively reduces particle pollution within the system through this priority-based control mechanism.

III. DESIGN PLATFORM

In order to achieve their objectives, various projects employ two distinct methodologies: hardware and software. Each methodology offers unique approaches and techniques to effectively accomplish project goals. The hardware methodology involves the design and implementation of physical components and systems, using tangible materials, electronic circuits, and mechanical structures. Collaboration between hardware development specialists, engineers, and technicians is crucial in creating, assembling, and testing these physical components. Tasks such as circuit board design, soldering, 3D printing, wiring, and integration of hardware components are key in the hardware methodology. It is vital for the project team to ensure the robustness, efficiency, and functionality of the hardware. Conversely, the software methodology focuses on developing and implementing software programs or algorithms to attain project objectives. This methodology entails coding, algorithm creation, and the utilization of software tools to solve specific problems or automate tasks. Software developers, programmers, and engineers specializing in software development collaborate to design, code, test, and refine the software components. Activities such as coding in programming languages, debugging, optimizing algorithms, and integrating software modules are involved in the software methodology. The project team must guarantee the reliability, scalability, and user-friendliness of the software to meet project requirements. The hardware and software methodologies possess their own strengths and weaknesses, rendering them suitable for different aspects of a project. The hardware methodology is indispensable for projects involving physical implementation or interaction with the physical world, while the software methodology is well-

suitable for projects centered around data processing, automation, or complex computations. In certain cases, projects may necessitate a combination of both methodologies, where hardware and software components work together to achieve the desired outcome. Ultimately, the selection between the hardware and software methodologies depends on the project's nature, objectives, and requirements. It is imperative for project teams to carefully assess needs and limitations in order to determine the most suitable methodology or combination thereof, ensuring successful project execution.

• General Hardware blocks & components

Figure 5 depicts the comprehensive block diagram of the Virtual Plant Monitoring System (VPMS). The VPMS comprises four main nodes, with one node serving as the master and the remaining three nodes functioning as slave nodes. The master node is connected to a personal computer (PC), while the slave nodes are equipped with wireless modules using Internet of Things (IoT) technology. Additionally, all three slave nodes are connected to temperature sensors.

The master node acts as the central hub of the VPMS, facilitating communication and data exchange between the PC and the slave nodes. It serves as the primary interface for the system, allowing users to interact with the VPMS through the PC. This connection enables users to monitor and control various aspects of the virtual plant being monitored. On the other hand, the slave nodes are responsible for collecting data from the physical environment. They are equipped with wireless modules, leveraging IoT technology, to establish wireless connections with the master node. These wireless modules enable seamless and efficient communication between the master and slave nodes, ensuring real-time data transmission. In addition to the wireless modules, each slave node is equipped with temperature sensors. These sensors are strategically placed within the virtual plant environment to capture temperature data at different locations. The sensors continuously measure the temperature and provide accurate readings to their respective slave nodes. These temperature readings are then transmitted to the master node via the wireless modules for further processing and analysis. Overall, the VPMS offers a comprehensive solution for monitoring and managing a virtual plant. The master node, connected to a PC, acts as the central control unit, while the slave nodes, equipped with wireless modules and temperature sensors, collect and transmit data to the master node. This system architecture enables efficient monitoring, control, and analysis of temperature-related parameters within the virtual plant environment.

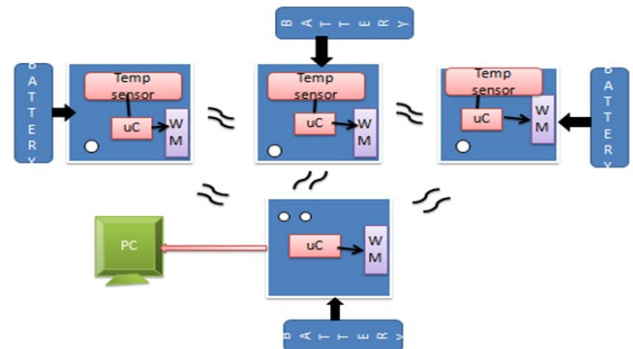


Figure. 5 General block diagram of VPMS

• Development Tools for dot net

The .NET framework is a comprehensive system that empowers developers to create a diverse range of applications, including console window-based, web-based, and mobile applications. It supports multiple programming languages like C#, VB.NET, and J#.NET. In our current project, we have opted to utilize the Visual Basic .NET (VB.NET) 2008 platform, which is a versatile programming language developed by Microsoft. VB.NET is designed with Rapid Application Development (RAD) in mind, providing developers with tools and features that significantly reduce the time required for application development. By leveraging VB.NET's capabilities, we can efficiently create software solutions that meet our specific needs. Its intuitive syntax, extensive library of pre-built functions and controls, and support for object-oriented programming principles simplify the development process, allowing us to create modular and scalable applications.

By using the VB.NET 2008 platform within the .NET framework, we can take advantage of its advancements, incorporating the latest features and improvements. This ensures that our application benefits from enhanced performance, security, and compatibility with modern technologies. In summary, utilizing the Visual Basic .NET 2008 platform enables us to harness the power and flexibility of this programming language to create efficient and robust applications. The RAD-focused design and extensive toolset provided by VB.NET streamline our development process, allowing us to deliver high-quality software solutions within a shorter timeframe.

When designing "form1" using VB.NET, various toolboxes are employed to enhance the form's visual appearance and functionality. These toolboxes include components such as the menu strip and group box. Each toolbox used in the design of "form1" has corresponding code generated and implemented at the backend of the form. In other words, the process of designing "form1" in VB.NET involves selecting and arranging different graphical elements from toolboxes like the menu strip and group box. These toolboxes provide

pre-built functionality and visual components that can be easily added to the form. Behind the scenes, the software automatically generates the necessary code to implement the selected toolboxes, ensuring the desired functionality is achieved when the form is executed. A flowchart is a visual representation of the steps or processes involved in a software program. It provides a graphical depiction of how the program progresses from one stage to another, illustrating the logical flow of control and the sequence of operations. The flowchart typically begins with a start symbol, indicating the program's initiation. Different symbols, such as rectangles, diamonds, and parallelograms, are used to represent various actions, decisions, and input/output operations within the software. Rectangles symbolize processes or tasks in the program, representing actions like calculations, data manipulation, or input/output

operations. Diamonds indicate decision points or conditional statements, where the program branches off in different directions based on specific conditions. Parallelograms represent input or output operations, signifying the interaction between the program and the user or external devices. Lines or arrows connect these symbols, indicating the flow of control and the logical order of execution. Connectors or off-page references may also be used to represent portions of the program located on different pages or sections of the flowchart, ensuring a comprehensive and organized representation of complex software programs. Overall, flowcharts provide a visual representation of a software's logic and structure, aiding in understanding the sequence of operations, identifying potential errors, and effectively communicating the program's functionality.

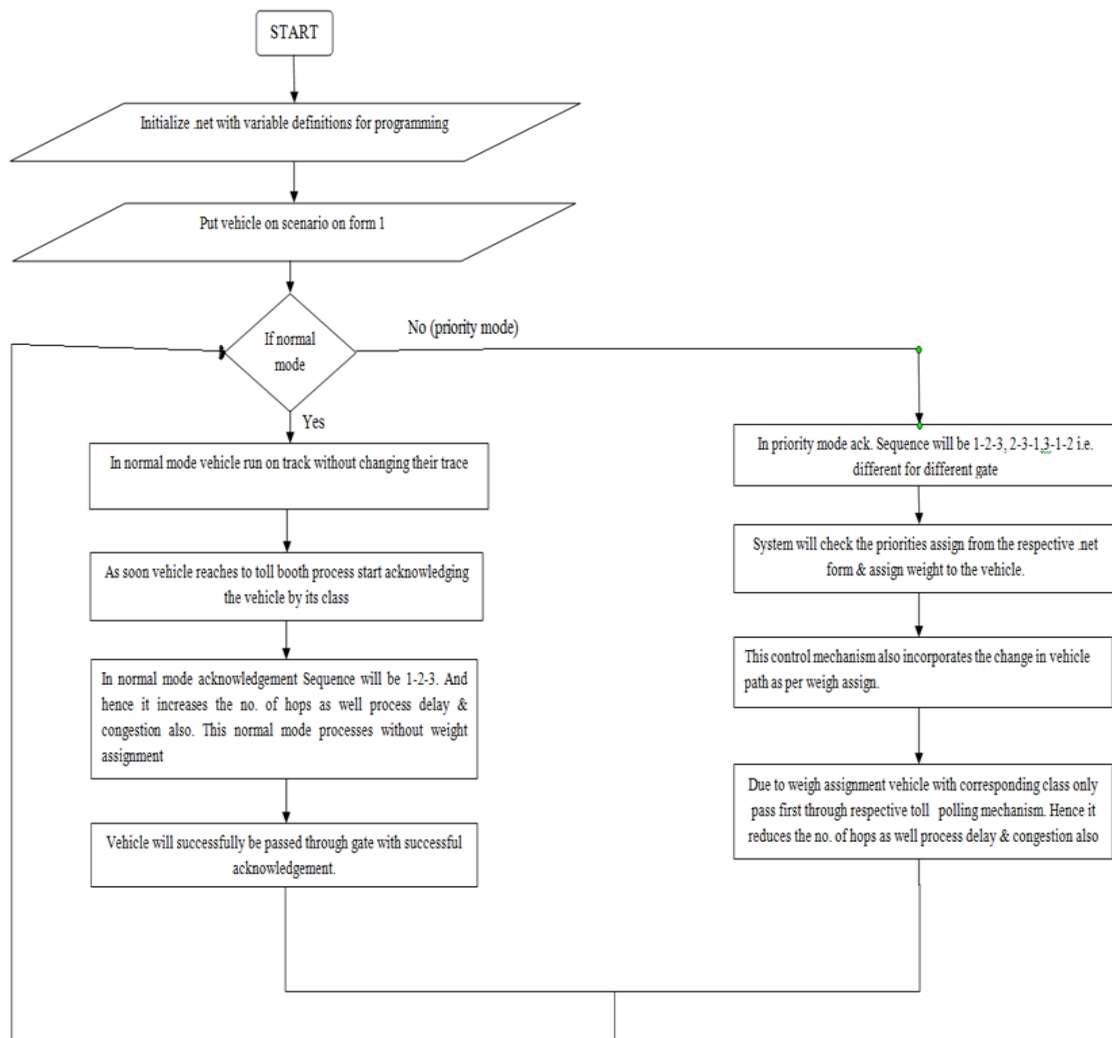


Figure.6 Design of form1 in VB.net

IV. TESTING-PLANNING &REPORTS

- **Hardware Testing**

The PCB layout process for motherboards and daughter cards comprises several key steps. Initially, the bare board undergoes a thorough inspection based on the layout diagram to confirm its proper manufacturing and absence of visible defects. Following the completion of the bare board inspection, the PCB is subjected to power supply testing and a smoke test. This entails supplying power to the board and examining voltage levels at different test points and IC pins to verify the power supply's integrity and identify any potential safety concerns. Moreover, the testing is carried out within specific time limits to ensure the board operates as intended. If any issues arise during the power supply and smoke test, further hardware verification becomes necessary. It is crucial to establish the correct functioning and proper connection of hardware components before proceeding with software testing. Without reliable hardware, effective software testing would be impossible. Once the hardware verification is complete, the PCB is tested for any minor connectivity problems that may exist. This step ensures the proper functioning of all PCB connections, including traces and solder joints, guaranteeing a defect-free and well-performing board.

In summary, the PCB layout process encompasses the inspection of the bare board, power supply and smoke testing, hardware verification, and testing for minor connectivity issues. These steps are essential to prepare the PCB for software testing, ensuring its overall functionality and reliability.

- **Software Testing**

Software testing can be approached in two main ways: automated testing and manual testing. Automated testing involves using specialized software tools to execute test cases and compare actual results with expected results. It saves time and is particularly useful for repetitive or complex tests. Tools like MPLAB, ICD2, and MPLAB SIM simulator are commonly used in automated testing to run tests, simulate scenarios, and analyze output. On the other hand, manual testing relies on human intervention to execute test cases and evaluate software behavior. While it can be time-consuming and prone to human error, manual testing offers a more exploratory approach, uncovering unexpected bugs and assessing user experience. Manual testing can also benefit from software debug tools like MPLAB, ICD2, and MPLAB SIM simulator for additional insights. When testing Vehicle Performance Monitoring System (VPMS) modules, the .NET framework and various tools and libraries are used for comprehensive testing. The software testing process includes steps such as code build, target image creation, memory compositions testing, optimization configuration testing, and protocol interoperation testing. Each step ensures the software's

functionality, performance, and adherence to desired specifications. By following these steps and employing appropriate tools, developers can identify and address software issues, ensuring the quality and reliability of the final product.

- **Application Testing**

The testing procedure involves evaluating the modules to ensure they satisfy the predetermined criteria and correspond to the original expectations stated in the project proposal. Furthermore, application testing is carried out to assess the performance of the application utilized for calculating results. It is vital to test the joining process of the IOT node before proceeding with any observations. This can be accomplished by observing the signal LED on the development boards. Once these preliminary tests are confirmed as successful, the project is implemented, and numerous readings are gathered for analyzing system interactions.

V. RESULTS & DISCUSSION

Figure 7 portrays a graph that showcases the correlation between travel time and the pollution index, focusing on two modes of transportation: normal mode and priority mode. The pollution index is represented on the x-axis, while the corresponding travel time is represented on the y-axis. The graph displays the outcomes for both modes, with the normal mode indicated by the blue line and the priority mode represented by the red line. A thorough examination of the graph reveals a consistent pattern: the travel time required for the priority mode consistently outperforms that of the normal mode across various pollution index values. This graph offers valuable insights into the influence of transportation mode on travel time concerning the pollution index. The priority mode emerges as a more efficient and time-saving alternative when compared to the normal mode. This information holds significance for decision-making processes concerning transportation planning, environmental considerations, and overall effectiveness in managing travel times in diverse scenarios.

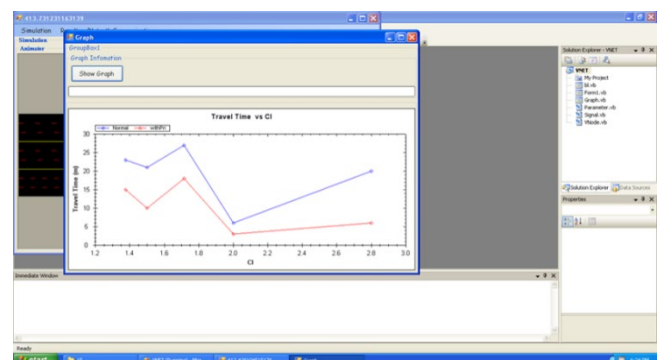


Figure 7.Travel time vs. Pollution index to reduce the time complexity

The graph in Figure 8 visually depicts the correlation between speed and the Pollution index, showcasing how adjusting these parameters can enhance the system's performance. Two lines are featured on the graph: a blue line representing normal mode speed and a red line representing priority mode speed. Analysis of the graph reveals that increasing the speed leads to a corresponding improvement in the system's performance. Higher speeds result in better functionality aligned with the system's purpose. It is important to consider the Pollution index, which signifies the system's emitted pollution levels during operation. Excessive pollution can detrimentally affect both the system's overall performance and the environment. The graph effectively demonstrates that by controlling the speed and Pollution index, it is possible to optimize system performance. The positions and trends of the blue and red lines on the graph illustrate the relationship between speed, Pollution index, and system performance. Adjusting these parameters achieves a balance, resulting in improved performance and reduced negative environmental impacts.

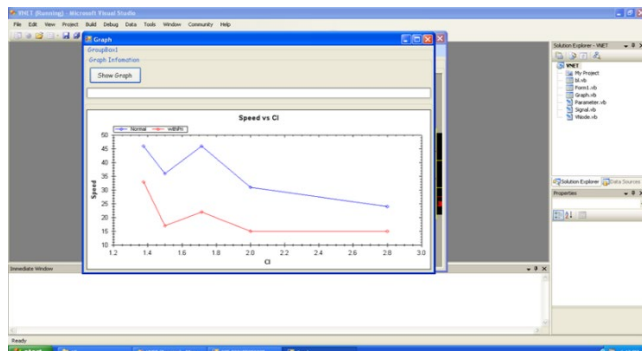


Figure 8. Improving the performance by controlling Speed and Pollution index

The graph presented in Figure 9 portrays the simulation end-to-end delay, representing the time it takes for a signal or data packet to travel from its source to the intended destination in a simulated scenario. This graph compares the normal mode with the priority mode in terms of the simulation end-to-end delay. Initially, the normal mode exhibits a higher delay, indicating a relatively longer time for the signal or data packet to reach its destination. However, when the priority mode is activated, the simulation end-to-end delay is significantly reduced or "suppressed." Essentially, the delay experienced in the priority mode is considerably less compared to that observed in the normal mode. This substantial reduction in simulation end-to-end delay emphasizes the positive impact of prioritizing specific signals or data packets on their delivery time. By prioritizing these critical signals or data packets, their timely transmission to the intended destinations is ensured with minimal delay. In summary,

Figure 6-3's graph vividly illustrates the advantageous effect of implementing the priority mode, demonstrating how it efficiently minimizes the simulation end-to-end delay and enhances the transmission process's efficiency and timeliness.

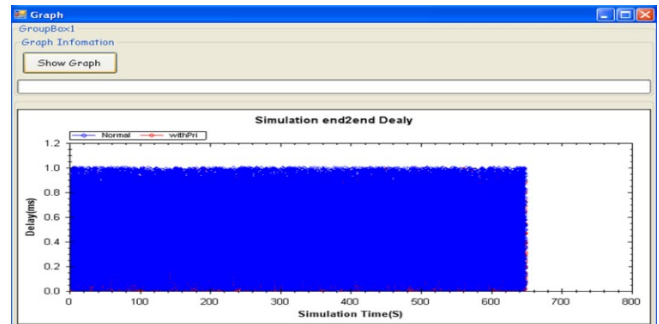


Figure 9. Simulation end to end delay

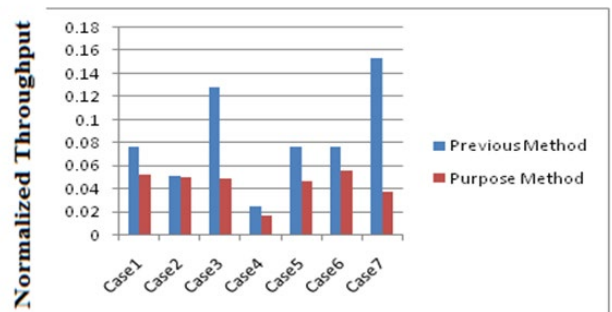


Figure 10. Normalized throughput

Figure 10 presents a graph comparing the normalized throughput of a specific method with a previous method. Normalized throughput refers to the amount of data processed or transmitted per unit of time, accounting for any variations between the methods. The graph clearly shows that the purpose method has a significantly lower normalized throughput compared to the previous method. This indicates that the purpose method is less efficient in handling data within the same time frame. The graph visually demonstrates the performance difference between the two methods, with the purpose method consistently exhibiting lower throughput values. It suggests that the previous method outperforms the purpose method in terms of data processing or transmission, handling a larger volume of data within the same time period. The difference in normalized throughput implies that the purpose method may have limitations or inefficiencies affecting its performance. Further analysis is necessary to identify the specific factors contributing to the lower throughput and explore areas for improvement to enhance the efficiency of the purpose method.



VI. CONCLUSION

PCCP, the Pollution Control Protocol, is a specialized upstream protocol designed specifically for VPMS, the Virtual Pollution Monitoring System. Extensive research and analysis have identified several important properties of PCCP. Firstly, it utilizes inter-arrival and service times of pollutants to accurately measure pollution levels at each sensor node, enabling a comprehensive understanding of pollution distribution within the network. PCCP also introduces a node priority index, allowing for weighted fairness implementation, where nodes with higher priority receive preferential treatment, ensuring equitable resource distribution. Simulation results have shown that PCCP achieves high link utilization, maximizing data transmission rates, and provides flexible fairness for balanced resource allocation. It effectively manages buffer size, reducing the loss of pollutants, thereby improving energy efficiency and throughput capabilities. PCCP offers desired throughput for different particle types, maintains high link utilization, manages queue length to minimize pollutant loss, and achieves a low pollutants drop rate. It considers memory requirements and configurations of mote devices, making it a feasible and practical solution. In conclusion, PCCP demonstrates potential as an effective protocol for pollution control in VPMS, offering accurate measurement, weighted fairness, high link utilization, reduced pollutants loss, energy efficiency, lower delay, and improved throughput capabilities. Its compatibility with existing mote configurations makes it a valuable option for integrating reliability mechanisms and enhancing fairness.

VII. REFERENCES

- [1]. S. Sridevi, M. Usha, and G. Pauline Amertha Lithurin, "Priority Based Pollution Control For Heterogeneous Traffic In Multipath Wireless Sensor Environments," in 2012 International Conference on Computer Moving and Informatics (ICCCI-2012), Jan. 10-12, 2012, Coimbatore, India.
- [2]. M. M. Monowar, M. O. Rahman, A. S. K. Pathan, and C. S. Hong, "Pollution Control Protocol for Wireless Sensor Environments Handling Prioritized Heterogeneous Traffic," in Proceedings of the 5th annual international conference on mobile and ubiquitous systems: computing, networking and services (ICST-MobiQuitous), ICST, Brussels, Belgium, 2008, pp. 1-8. doi: 10.4108/ICST.MOBIQUITOUS2008.3806
- [3]. C. T. Ee and R. Bajcsy, "Pollution control and fairness for many-to-one environments routing in Sensor," in Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), Nov. 2004. doi: 10.1145/1031495.1031514
- [4]. M. M. Monowar, M. O. Rahman, and C. S. Hong, "Multipath Pollution control for heterogeneous traffic in wireless sensor environments," in Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), 2008. doi: 10.1145/1460412.1460435
- [5]. M. Y. Darus and K. A. Bakar, "Pollution Control Algorithm in VPMSs," World Applied Sciences Journal, vol. 21, no. 7, pp. 1057-1061, 2013. ISSN: 1818-4952
- [6]. C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Pollution detection and avoidance in sensor environments," in Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), Nov. 2003. doi: 10.1145/958491.958497
- [7]. C. Wang, M. Daneshmand, and B. Li, "A Survey of Transport Protocols for Wireless Sensor Environments," in Proceedings of IEEE INFOCOM - IEEE Conference on Computer Communications Workshops, May-June 2006, Arkansas, USA.
- [8]. B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Pollution in wireless sensor environments," in Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), Nov. 2004. doi: 10.1145/1031495.1031513
- [9]. D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly resilient, energy-efficient multipath routing in wireless sensor environments," ACM Mobile Computing and Communications Review, vol. 1, no. 2, pp. 10-24, 2002. doi: 10.1145/505586.505588
- [10]. M. H. Yaghmaee and D. A. Adjeroh, "A new priority based Pollution control protocol for wireless multimedia sensor environments," in Proceedings of the 9th ACM International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (WOWMOM), June 2008, pp. 1-8. doi: 10.1145/1409944.1409951
- [11]. L. Wischhof and H. Rohling, "Pollution control in Vehicular Pollution Management Systems," in IEEE International Conference on Vehicular Electronics and Safety, Oct. 2005.
- [12]. D. Sutariya and S. Pradhan, "An Improved AODV Routing Protocol for VANETs in City Scenarios," in IEEE International Conference on Advances in Engineering, Science, and Management (ICAESM), March 2012.
- [13]. M. H. Yaghmaee and D. A. Adjeroh, "A new priority based Pollution control protocol for wireless multimedia sensor environments," in Proceedings of the 9th ACM International



Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (WOWMOM), June 2008.

- [14]. [14] S. A. Kahurke and B. N. Mahajan, "Implementation of Priority Based Scheduling and Pollution Control Protocol in Multipath Multi-Hop WSN," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 1, no. 2, June 2012. ISSN: 2277-3878
- [15]. [15] I. Howitt, "IoT Performance in the Presence of 802.11b WLAN," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, Nov. 2002. doi: 10.1109/TVT.2002.805331
- [16]. [16] S. M. Mahmud and S. Shanker, "In-Vehicle Secure Wireless Personal Area Environment (SWPAN)," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, May 2006. doi: 10.1109/TVT.2006.875874
- [17]. [17] P. Bhagwat, "IoT: technology for short-range wireless apps," in *Proceedings of IEEE Computer Society*, May-June 2001.