A Survey on Wireless Sensor Networks with Air Pollution Dispersion Modeling

Tansen Patel¹, P Udayakumar²
¹Department of Computer Science & Engineering, SSIPMT, Raipur (CG), India
²Department of Computer Science & Engineering, CCEM, Raipur (CG), India

Abstract— Wireless Sensor Networks are very popular technology to monitor environmental conditions consists of distributed independent sensors. Air pollution in heavily populated and industrialized areas is a serious problem. WSNs prepared with different sensors have been actively used for air quality monitoring. The air pollution monitoring system contains sensors to monitor the concerned pollution parameters in environment. In this paper, several technologies of a WSN based air pollution monitoring system and air quality model for air pollutions have been discussed. Various types of air pollutants with its effects on health and environment are discussed initially. It includes an overview of operating systems and simulators specifically designed for WSNs with various design approaches, and investigation of the current wireless sensor motes concerning towards performance metrics. Finally, this study reviews existing atmospheric dispersion model specifically the Gaussian plume model and its equations for air pollution.

Keywords— Wireless Sensor Networks; air pollution; operating system; simulator; dispersion model

I. INTRODUCTION

In recent years, wireless sensor networks (WSNs) have achieved global interest particularly with the rise in Micro-Electro-Mechanical Systems technology which has facilitated the growth of smart sensors. Sensors are small, inexpensive, less processing and computing resources. Sensor nodes can sense, compute, and collect information from the surroundings and transmit the sensed data to the system. Smart sensor nodes organized with sensors, processor, memory, power supply, radio, and an actuator [1]. Environmental pollution includes air pollution, water pollution and soil pollution are now global problem in most of the countries.

The major gases that reason of air pollution are nitrogen oxides NOx (NO and NO2), carbon monoxide (CO), sulfur oxides (SOx), ozone (O3) emitted from automobile and coal fired plants [2]. To monitor and control the air pollution is very necessary and best approach to control air pollution is monitor exceeding levels of air pollutants and it must be detected and reduced. Environmental monitoring has generated a large amount of information collected by sensors, which are generally measuring temperature, humidity or air pressure [3]. A wireless sensor network configures numbers of sensors required to recognize about the growth and use of low cost sensors that are connected to wirelessly networked nodes [4]. The application air pollution monitoring system includes the GSM, GPRS, etc. however wireless nodes installation and maintenance are costly and have been rapidly developed in wireless sensor network. To reduce the cost and size of the sensors are developing by many researchers in the world for air pollution monitoring system [5]. The main aim of dispersion model is to present ambient first-level concentrations of emitted substance given information about the emissions gases in the atmosphere. The amount of emission can be determined from level of the industrial process. The quality of air refers to the concentration in the immediate surroundings in the atmosphere, not in the emission source. This can be determined in the ground-level concentrations that may arise at various distances from the source [6].
The remainder of this paper is organized as follows. Section II, review some research activity on air pollution monitoring system using WSN technologies. Section III, we illustrate air pollutants, operating systems, simulators for WSNs, and sensor motes of a wireless sensor node platform. Section IV discuss on dispersion modeling techniques in the air pollutants. Discussion of this paper to be used in Section V. The conclusion and future application have been given in section VI.

II. LITERATURE SURVEY

Many research activities are undergoing on air pollution monitoring system. A Mobile Air Quality Monitoring Network (MAQUMON) that used in a large area to moving vehicles ready with sensor nodes to monitor air quality [7]. Every sensor node consists of a microcontroller, sensors to sense the concentrations of nitrogen dioxide, Carbon Monoxide, and ozone, and an on-board Global Positioning System unit. The node was able to send the sensed data to the gateway through the Bluetooth connection in a car. When the car moves, the sensor nodes sense the concentrations every minute and collect the data marking with location information into a memory.

An integrated mobile environmental sensing system [8] to support the management of transport and urban air quality is used. Sensor nodes are deployed to monitor traffic, weather and pollutant concentrations on vehicles and infrastructure, and send data into a dynamically configurable computing platform that supports both longer term strategic planning decisions and near real-time event management.

Wireless Sensor Network Air Pollution Monitoring System (WAPMS) [9] to monitor air pollution in Mauritius is used through the use of large number of wireless sensors deployed around the island. In order to improve the efficiency of WAPMS, Recursive Converging Quartiles (RCQ) designed and implemented a data aggregation algorithm. The algorithm which significantly reduces the amount of data to be transmitted to the sink and saving energy is used to filter out invalid readings, merge data to eliminate duplicates, and summarize them into a simpler form. A hierarchical routing protocol for better power management used which causes the nodes to sleep during idle time.

An Indoor Air Quality monitoring (IAQ) system [10] that integrates a power management approach using a sensor network to decrease sensors energy consumption for metal oxide semiconductor (MOX) gas sensors by using an adaptive duty cycling mechanism. A Wireless Sensor Network (WSN)-based urban air quality monitoring system [11] that is connected to a GSM based centralized control system by a LabVIEW program that supplies sensed data in a database. It is implemented to monitor the carbon monoxide concentration that is the monitoring systems in the city road of Taipei for vehicle emissions. In 2011, an air quality monitoring system proposed based on ZigBee wireless sensing technology [12] to observe with requirements of oil and gas industry. It uses ZigBee wireless network to send outcomes to the monitoring center. A quick warning will be generated if some abnormal situations happen, to remind staff to get effective actions to avoid major accidents and protect human lives in industry.

In 2012, a wireless sensor network to monitor air pollution levels of different pollutants due to environment changes [13] is used. This system proposed a method which typically focuses on longer sustain time period of sensor network by effectively organizing energy in sensor network, efficiently processing of stored information and less overhead in moving information between different sensor nodes.

III. WSNs FOR AIR POLLUTION MONITORING SYSTEM

A. Air pollutants

Every year increasing growth of industries and vehicles traffic have affected environment. Air pollution is the materials of particulate matter, biological materials or chemicals, which are the reason to harm humans or other living organisms, or reason to damage the natural environment or built environment, into the atmosphere. There are some air pollutants gas effects the quality of air.
### Table I. Effects of air pollutants

<table>
<thead>
<tr>
<th>Air Pollutants</th>
<th>Health Effect</th>
<th>Environment Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>Flu-symptoms, Heart dieses, Chest pain,</td>
<td>Biomass burning, affects on greenhouse gases</td>
</tr>
<tr>
<td>NO₂</td>
<td>Asthma, Emphysema, Thyroid gland, Bronchitis disease</td>
<td>Terrestrial ecosystem, Destruction of cilia, Injuries to plants</td>
</tr>
<tr>
<td>O₃</td>
<td>Coughing, Throat irritation, Asthma, Chest pain, Affect on lung function</td>
<td>Appearance of leaf, Affects on crop yields, and forest growth</td>
</tr>
<tr>
<td>SO₂</td>
<td>Lung cancer, irritation of eyes and skin, asthma, respiratory problem</td>
<td>Acid rain, Damage of plant and water, Aesthetic damage</td>
</tr>
<tr>
<td>PM</td>
<td>Heart and lung disease, Cause of asthma, Bronchitis, pneumonia</td>
<td>Change in nutrient cycle, Acid deposition of air and water, Affects on ecosystem</td>
</tr>
<tr>
<td>Pb</td>
<td>Effect on bones and kidney function, Affects nervous system</td>
<td>Poisoning to soil organism, Intoxication disease</td>
</tr>
</tbody>
</table>

### B. Operating system for WSNs

**Table II. Overview of OS for WSN**

TinyOS [14], developed in UC Berkeley, a flexible design and low resource utilization, programming is based on mechanism which are wired collectively to form an application at design time. Contiki [15], supports dynamically loadable modules. To deal with the programming difficulty of event-based programming, Contiki supports cooperative multi-threading. SOS [16], supports dynamically loadable modules, adopts a module-based architecture and support dynamic memory allocation. To enable full-fledged multi-threading support, Mantis OS [17] implements a conventional preemptive time-sliced multi-threading on sensor nodes. To facilitate the conventional
programming paradigm Mantis kernel also supports synchronous I/O and a set of concurrency control primitives. Time-sensitive WSN applications surveillance and environmental monitoring supported by Nano-RK [18], implements a reservation-based real-time OS for WSNs. Nano-RK for guaranteeing that task deadlines are met to supports fixed-priority based preemptive multitasking. CPU and network bandwidth reservations are also supports, such as tasks can specify the OS provides timely and their resource demands, guaranteed and controlled access to network bandwidth and CPU cycles. RETOS [19], is designed to improve several aspects of prior work. It improves system flexibility by supporting dual mode operation as well as application code checking at design-time and run-time. It also supports loadable modules and provides multi-hop networking services. The most recent version of RETOS, version 1.4, supports a wide range of platform, optimizes the networking layer, and improves system safety.

LiteOS [20], is designed to present a conventional Unix-like environment for programming WSN applications. It contains a wireless shell for user interaction and a built-in hierarchical file system using Unix-like commands, kernel support multithreaded applications for dynamic loading native execution and an object-oriented programming language that uses a subset of C++ as its syntax with class library support. Enix [21] OS supports dynamic loading using kernel supported PIC, which is simple to port to other platforms. The dynamic library is pre-linked to the kernel with minor modification. Hence, our runtime additional buffer and overhead are reduced compared to the runtime relocation approach.

C. Simulators for WSNs

<table>
<thead>
<tr>
<th>WSNs simulators</th>
<th>Programming language</th>
<th>GUI</th>
<th>Open Source</th>
<th>Features</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SensorSim</td>
<td>C++</td>
<td>No</td>
<td>Yes</td>
<td>Power and communication protocol models Sensing channel and sensor models, Scenario generation, Support for hybrid simulations</td>
<td>Limited in SensorSim project realism, Consider limited resources of sensor nodes, Simulates the complete WSN protocol stack</td>
</tr>
<tr>
<td>TOSSIM</td>
<td>C++</td>
<td>Yes</td>
<td>Yes</td>
<td>Can be targeted to motes without modification, Nodes share the identical code image, The develop algorithms can be tested on a output platform</td>
<td>Makes several assumptions about the target hardware platform, making some behavior accurate while simplifying others</td>
</tr>
<tr>
<td>TOSSF</td>
<td>C++</td>
<td>Yes</td>
<td>Yes</td>
<td>Primary focus on scalability, Support heterogeneous nodes and dynamic topology</td>
<td>Long test-debug cycles</td>
</tr>
<tr>
<td>Emstar</td>
<td>C</td>
<td>Yes</td>
<td>Yes</td>
<td>Supports hybrid form, Presented a choice to interface with actual hardware while running a simulation, Compatible with two different types of node hardware</td>
<td>Supports only the code for the types of nodes that it is designed to work with</td>
</tr>
<tr>
<td>SENS</td>
<td>C++</td>
<td>No</td>
<td>Yes</td>
<td>Multiple different component implementations</td>
<td>Less customizable, Only measurable phenomenon is sound</td>
</tr>
<tr>
<td>J-sim</td>
<td>Java</td>
<td>Yes</td>
<td>Yes</td>
<td>Ability to simulate the use of sensors for phenomena detection,</td>
<td>Comparatively complicated to use,</td>
</tr>
</tbody>
</table>
### Table III. Overview of simulators for WSN

<table>
<thead>
<tr>
<th>Simulator</th>
<th>Programming Language</th>
<th>Support for using the simulation code for real hardware sensors</th>
<th>Unnecessary overhead in the intercommunication model</th>
<th>Full functionality of a programming language can be used, Option to split the visualization from the simulation</th>
<th>Does not directly support sensor networks at the physical layer</th>
<th>Incomplete nature of the tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dingo</td>
<td>Python</td>
<td>Yes</td>
<td>Yes</td>
<td>Full functionality of a programming language can be used, Option to split the visualization from the simulation</td>
<td>Does not directly support sensor networks at the physical layer</td>
<td>Incomplete nature of the tool</td>
</tr>
<tr>
<td>Shawn</td>
<td>C++</td>
<td>No</td>
<td>Yes</td>
<td>Able to simulate large-scale WSNs, Ability of selecting the application preferred behavior, Full access to the communication graph</td>
<td>Visualization output does not support, MAC module is not extent, Lots of programming is required</td>
<td></td>
</tr>
<tr>
<td>Castalia</td>
<td>C++</td>
<td>Yes</td>
<td>Yes</td>
<td>Physical process modeling, sensing device bias and noise, Node clock drift, Several MAC and routing protocols executed, Highly tunable MAC protocol and flexible parametric physical process model</td>
<td>Not useful if test code compiled for a specific sensor node platform, Not a sensor specific platform</td>
<td></td>
</tr>
</tbody>
</table>

The Wireless Sensor Network applications require an end-to-end data transmission to achieve the required performance. It requires much effort and is expensive to study and analyze the behaviors of such WSNs by means of a test bed or deploying WSNs.

SensorSim [22] builds on the NS-2 simulator providing additional capabilities for modeling WSNs. SensorSim simulates the whole WSN protocol stack, even though this can be observed as overkill and adding redundant complexity as this is not required in order to simulate the expected behavior, formulates the SensorSim platform complex to use. TOSSIM [23] is platforms specifically designed to simulate WSNs, which is a part of the TinyOS development efforts and discrete-event simulator for TinyOS applications. It aims to development and debugging by compiling applications into the TOSSIM framework to support TinyOS application, which runs on a PC instead of compiling for a mote. Programs can be directly targeted to motes without modification used by the TOSSIM framework. This provides users a better margin to analyze, test, and debug algorithms in a controlled and repeatable environment. TOSSF [24] is the stateless wireless adhoc networks simulation framework that compiles a TinyOS application. Although it enables development of tradition environmental models, the absence of a scripting framework needs those models to be compiled into the simulation framework. This simulators is tightly coupled with TinyOS, may be unsuitable for early prototyping, or developing portable WSN applications.

EmStar [25] is a component based, discrete-event framework that present a range of run-time environments, from pure simulation, distributed deployment on iPAQs. Its component based design allows for fair scalability. The main goal of Emstar is to reduce design simplify and accelerate the design of new sensor network complexity, enabling work to be shared and reused, and to applications. SENS [26] is a customizable component-based simulator for WSN applications consists of network communication, the physical environment and interchangeable and extensible components for applications. In SENS, each node is partitioned into key components such as an application that simulates the software and environment that is used for network propagation characteristics. SENS defines network models for successfully forwards packets to all neighbors, delivers with a possibility of loss based on a fixed probability, and consider the probability of collision at each node.

J-Sim [27] is a component-based discrete event simulator after NS-2 modeled in Java uses the concept of components instead of the concept of having an object for each individual node. J-Sim uses level components the target node which creates stimuli, the sensor node that respond to the
stimuli, and the sink node which is the final destination for stimuli reporting. Each component is scattered and modeled differently within the simulator to use different protocols in different simulation runs. Dingo [28] provides a workbench for prototyping algorithms for WSNs taking a top-down design methodology. Dingo consists of a fixed API, with customizable internals, a simple GUI and a set of base classes. Simulated sensor nodes run in threads and communicate using the same protocols. Sensors are modeled using a pool of synchronized, communicating threads. Shawn simulates large-scale WSNs, where physically correct simulations fail. The design behind Shawn is to use conceptual models to simulate the concern of a phenomenon rather than the phenomenon itself [29]. Castalia is a network simulator of low-power embedded devices for Wireless Sensor Networks based on the OMNeT++ platform. Researchers and developers who want to analyze their distributed algorithms or protocols in realistic radio models and wireless channel, with a realistic node performance especially relating to access of the radio. Castalia can also be used to calculate in various platform characteristics for specific applications with highly parametric and simulate a wide range of platforms [30].

D. Sensor Motes

Wireless sensor network motes are the small, low power single board computer for wireless communication with a radio to collect and transfer data. Currently commercially available mote platforms are to be presented, Mica2/Micaz [31] from CrossBow Technology are the second and third generation mote technologies. TelosB/Tmote Sky [32] are developed by University of California, Berkeley and presently available from MEMSIC Inc. SHIMMER [33] the Intelligence Sensing Health, Experimental Reusability, Modularity, and Mobility designed for wearable health sensing in mutually connected with wireless environments by Real-time Technologies. IRIS is the most modern motes from Crossbow Technologies. Sun SPOT [34] from Sun Microsystems are small Programmable Object Technology. EZ430-RF2480/F2500T [35] from Texas Instruments, which is support wireless networking solutions. Waspmotes [36] from Libelium Communications are open source hardware and software sensor platform.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>MICA2/MICAZ</th>
<th>TelosB/TmoteSky</th>
<th>Sunspot</th>
<th>SHIMMER</th>
<th>IRIS</th>
<th>eZ430F2500T</th>
<th>Waspmotes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Crossbow</td>
<td>UC Berkeley</td>
<td>Sun</td>
<td>Intel</td>
<td>Crossbow</td>
<td>Texas Instruments</td>
<td>Libenium</td>
</tr>
<tr>
<td>FLASH (Bytes)</td>
<td>128k</td>
<td>48k</td>
<td>4M</td>
<td>48k</td>
<td>640k</td>
<td>32k</td>
<td>128k</td>
</tr>
<tr>
<td>RAM (Bytes)</td>
<td>4k</td>
<td>10k</td>
<td>512k</td>
<td>10k</td>
<td>8k</td>
<td>1k</td>
<td>8k</td>
</tr>
<tr>
<td>EEPROM (Bytes)</td>
<td>512k</td>
<td>1M</td>
<td>-</td>
<td>-</td>
<td>4k</td>
<td>-</td>
<td>4k</td>
</tr>
<tr>
<td>OS/Programming/IDE support</td>
<td>TOS, Mantis</td>
<td>TOS, Mantis, Contiki, CCS, IAR,</td>
<td>J2ME, JDK, Squawk VM</td>
<td>TOS</td>
<td>MoteRunner, TOS, Motework</td>
<td>TOS, CCS, IAR</td>
<td>C++, waspmotes</td>
</tr>
<tr>
<td>Modulation Technique</td>
<td>O-QPSK</td>
<td>O-QPSK</td>
<td>O-QPSK</td>
<td>O-QPSK</td>
<td>O-QPSK</td>
<td>O-QPSK</td>
<td>PWM</td>
</tr>
<tr>
<td>Sensor support</td>
<td>Temperature, Humidity,</td>
<td>Temperature, Humidity,</td>
<td>Temperature, Light,</td>
<td>Accelometer</td>
<td>Light</td>
<td>Temperature, Humidity</td>
<td>Accelerometer, Temperature</td>
</tr>
</tbody>
</table>
IV. DESIGN AND ANALYSIS

A. Atmospheric dispersion modeling
An atmospheric dispersion model is a mathematical simulation to determine air pollutants disperse in the ambient atmosphere. The mathematical equations and algorithms which calculate pollutant dispersion is executed through computer programs. The dispersion models are used to calculate approximately the downwind ambient concentration of air pollutants release cause from vehicular traffic, accidental chemical releases, and expected concentrations under specific scenario, the dominant type of model used in air quality policy making is also used and for pollutants that are dispersed over large distances that may react in the atmosphere, a very high spatio-temporal variability and for epidemiological studies statistical land-use regression models for pollutants are also used. Atmospheric air quality dispersion models are generally used to estimate reduction quantities occurred during the carry pollutant from an industrial source and projects of pollution concentration are introduce at ground level. Dispersion models typically incorporate physical, meteorological, terrain, and chemical characteristics of the effluent and source intend to simulate the configuration and transport of pollutant plumes. A strict environmental instruction worldwide is following the growing concern about the reliability and validity of air quality dispersion models [37].

B. Air pollutant dispersion equations
A computer programs for estimating the dispersion of air pollutant emissions developed called air dispersion models. Gaussian plume model can be used to find realistic explanation of dispersion, represents methodical solution to the diffusion equation for defined conditions. The Gaussian plume dispersion model is treated as spreading outward from the centerline of the plume subsequent a normal statistical distribution in both the horizontal and vertical directions. The experiment involve two main stages first, the height to which the plume rises at a given downwind distance from the plume source is calculated. Next, the ground-level pollutant concentration based on the plume at the given downwind is predicted using the Gaussian dispersion equation, as [38] shown below:

$$G = \frac{S}{p} \frac{C}{\sigma_x \sqrt{2\pi}} \frac{v_1 + v_2 + v_3}{\sigma_y \sqrt{2\pi}}$$

Where G is concentration of emissions, g/m3, at any receptor located: w meters downwind from the emission source point, x meters crosswind from the emission plume centerline and y meters above ground level. S is the source pollutant emission rate, in g/s, p is the horizontal wind velocity along

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Light</th>
<th>Light</th>
<th>Accelero</th>
<th>, Light</th>
<th>ure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion connectors for attaching external sensor</td>
<td>Low power microcontroller and RF module</td>
<td>Open source hardware and software support</td>
<td>Support Bluetooth, Realtime capability, For long term wearable use</td>
<td>3-times radio range compared to MICA nodes at half the power consumption</td>
<td>Lowest cost, size</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Limitations</th>
<th>Hardware concentrated</th>
<th>Software optimization ignored in design</th>
<th>Expensive</th>
<th>Less number on board sensors</th>
<th>Work best with simpcI RT protocol</th>
<th>Complex hardware</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Light</th>
<th>Light</th>
<th>Accelero</th>
<th>, Light</th>
<th>ure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion connectors for attaching external sensor</td>
<td>Low power microcontroller and RF module</td>
<td>Open source hardware and software support</td>
<td>Support Bluetooth, Realtime capability, For long term wearable use</td>
<td>3-times radio range compared to MICA nodes at half the power consumption</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hardware concentrated</th>
<th>Software optimization ignored in design</th>
<th>Expensive</th>
<th>Less number on board sensors</th>
<th>Work best with simpcI RT protocol</th>
<th>Complex hardware</th>
</tr>
</thead>
</table>
The plume centerline, m/s, E is the height of emission plume centerline above ground level, in m, \( \sigma_x \) is horizontal standard deviation of the emission distribution, in m, \( \sigma_y \) is vertical standard deviation of the emission distribution, in m, H is the height from ground level to bottom of the inversion aloft, in m.

The crosswind dispersion parameter \( C = e^{\left(-x^2/(2\sigma_x^2)\right)} \)  
Vertical dispersion parameter \( V = v_1+v_2+v_3 \)
Where \( v_1 \) is the vertical dispersion with no reflections and \( v_2 \) is the vertical dispersion for reflection from the ground and \( v_3 \) is the vertical dispersion for reflection from an inversion aloft and computed as

\[
v_3 = \sum_{m=1}^{\infty} \left\{ e^{\left(-y-E-2mH)/(2\sigma_y^2)\right)} + e^{\left(-y+E+2mH)/(2\sigma_y^2)\right)} + e^{\left(-y-E+2mH)/(2\sigma_y^2)\right)} + e^{\left(-y+E-2mH)/(2\sigma_y^2)\right)} \right\}
\]

The above equation includes upward reflection from the ground and downward reflection from the bottom of any inversion lid present in the atmosphere. The sum of the four exponential terms in \( v_3 \) converges to a final value quickly. For most cases, the summation of the series with \( m = 1 \), \( m = 2 \) and \( m = 3 \) will give an adequate solution. \( \sigma_y \) and \( \sigma_x \) are functions of the atmospheric stability class and of the downwind distance to the receptor. The two most significant variables affecting the degree of pollutant emission dispersion achieved are the height of the emission source point and the amount of atmospheric turbulence. The more turbulence, the enhanced the degree of dispersion. The Gaussian air pollutant dispersion equation need the input of \( S \) which is the pollutant plume's centerline height above ground level and \( S \) is the sum of \( S_a \) an actual physical height of the pollutant plume's emission source point plus \( \Delta S \) that is the plume rise due the plume's buoyancy.

In Fig. 1 \( S_a \) is the Actual stack height, \( S_e \) is the Effective stack height and it represent pollutant release height

\[
S_e = S_a + \Delta S
\]

and \( \Delta S \) calculate the plume rise

Briggs separated air pollution plumes into general categories as Hot buoyant plumes in calm ambient air conditions, Hot buoyant plumes in windy ambient air conditions, Cold jet plumes in calm ambient air conditions, and Cold jet plumes in windy ambient air conditions.

A logic diagram for using the Briggs equations [39] to find the plume rise trajectory of bent-over buoyant plumes is presented below, in fig. 2, the Briggs equations parameters are used where the parameter \( \Delta S \) calculate the plume rise, in m, \( B_f \) is the buoyancy factor, in m³s⁻³, \( d \) obtain the downwind distance from plume source, in m, \( d_{max} \) obtain the downwind distance from plume source to point of maximum plume rise, in m, \( w \) is the windspeed at actual stack height, in m/s, and \( p \) is the stability parameter, in s⁻².

**Fig.1. Visualization of a buoyant Gaussian air pollutant dispersion plume**
Fig. 2. Logic diagram for Briggs equations to calculate the rise of a buoyant plume

V. DISCUSSIONS

Various features like issues, design characteristics and software in particular to OS motivate the design principles to be considered while designing an OS for WSN. The classification framework discussed features of a typical WSN OS and categorizes existing operating systems according to its application. To choose an operating system, analyzing different applications with their characteristics and suggesting ideal OS to an application developer. Simulation is an essential tool due to the unfeasible analysis and experimental complexity. Simulation model for a WSN presents guidelines to assist, selecting an appropriate and the most used available tools. A complete description of different simulators is valuable and allows choosing the one most appropriate for testing. A comparative study of performance metrics of existing and popular wireless sensor motes focusing issues like cost, size and power and need develop more potential sensor nodes. The outputs for air pollutant concentrations are often expressed as contour map regulate to demonstrate the spatial variation in contaminant levels. The contour lines can overlay sensitive receptor positions and disclose the spatial relationship of air pollutants to concern area.

While previous models rely on stability classes for the resolves $\sigma_x$ and $\sigma_y$ new modern models increasingly rely on the Monin-Obukhov similarity theory to derive parameters. Briggs measured the trajectory of cold jet plumes to be concern by initial velocity momentum, and the trajectory of hot, buoyant plumes to be dominated by buoyant momentum to be extended. Briggs's equations for bent-over, hot buoyant plumes are based on observations and information concerning plumes from representative combustion sources such as the flue gas stacks from steam-generating boilers burning fossil fuels in large power plants.

VI. CONCLUSION AND FUTURESCOPE

Air pollution involve spate of pollutants which produce a lot of chronic and sensitive diseases in human and affect the environment so the particles have to within the control of standard limit. The
survey presents existing design approaches of OS for WSN as well as explores new requirements to be used in the future applications of WSN. The survey of this paper has been to present comprehensive study and analysis on various types of sensor network simulators and the features and limitations of each simulator. Now it’s choose best existing motes today for required applications after studying all the aspect of sensor nodes and to work in executing the target. The discussed air pollutant dispersion models are useful to recognize the pollutants concentration, and the chemical analysis of the pollutants stored in the area through proper sampling is necessary to know the chemical characteristics.

REFERENCES
20. Q. Cao, T. Abdelzaher, J. Stankovic, and T. He, ”The LiteOS operating system: Towards unix-like abstractions for wireless sensor networks”, In proc. of the 7th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN), 2008.