Article Citation Format
Ezea, I.L. (2020): An Agent Based Smart Environment System for Wireless Surveillance and Intrusion Detection Journal of Digital Innovations & Contemp Res. In Science., Engineering & Technology. Vol. 8, No. 2. Pp 141-154
DOI: dx.doi.org/10.22624/AIMS/DIGITAL/V8N2P9

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 15th March, 2020
Review Type: Blind Final
Acceptance: 14th June, 2020
CrossREF Member Listing :: ttps://www.crossref.org/06members/50go-live.html

An Agent Based Smart Environment System for Wireless Surveillance and Intrusion Detection

Ezea, I. L.
Department of Math/Computer Science/Statistics/Informatics
Federal University
Ndufu Alike, Ikwo, Ebonyi State, Nigeria
E-mail: ezeaikenna@yahoo.com

ABSTRACT

Nigerian government has suffered great economic loss due to pipeline vandalism by Niger Delta militants. This issue of pipeline vandalism has been on for over 30 years. The government has tried several means to contain this problem, yet no solution seems at sight. The rehabilitation of the ex-militants and the oil surveillance contract awarded to some companies cost Nigerian government billions of naira. Oil spillage runs for weeks if not months causing environmental pollution and shortage in oil production. Most people have lost their lives, aquatic lives have been wasted, farmlands turned barren and above all Nigeria economy is in a very bad shape due to millions of crude oil barrels that are wasted on a daily basis. This problem is very weighty not only to the government but particularly to the people living in the region, since majority of them are farmers and their livelihood depend on aquatic lives and soil produce. This work addressed the issue of pipeline vandalism by providing wireless sensor network surveillance in the oil field and reporting any suspicious movement to the base station and to other designated authorities using text messages and virtual reality system. The approach ensures that the battery life is extended and that the cost of deployment is highly reduced.

Keywords: Agent, Smart Environment, System, Wireless Surveillance, Intrusion Detection

1. INTRODUCTION

The Niger delta region is one of the richest oil producing areas in Nigeria and the largest wetland in Africa. The region is estimated to have 37 billion barrels (bb) of oil reserves and 168 trillion cubic feet of gas deposits (Omotala, 2016). The oil sector account to over 90% of foreign exchange earnings for Nigeria and the bulk of the oil comes from the Niger delta region of Nigeria. Over the years, crude oil theft and spillage through pipeline vandalism is considered one of the major problems of the region. Rising cases of pipeline vandalism by militant groups have significantly affected oil production and the revenue base of the government and oil companies operating in the region. In spite of all the efforts made by the government to curb these activities, it continues unabated. In the government's quest to curb these crises, it has deployed several means: starting from the use of military intelligence, establishment of civil defense corps, the use of police, dialogue with the elders of the region, establishment of ministry of Niger Delta, etc. In spite of all these efforts the activities of the Niger Delta Militants is still making headlines on the Nigerian dailies.

These activities seems to have defiled every human efforts, because of the delicate nature of the environment and compromising and dubious nature of some human beings it may not be ideal to rely on people as a means of surveillance. Most environment are located in the forest, which, may be filled with wild animals, and some people may want to collect bribe in other to compromise the security of the environment. In other to put an end to these crises, an Agent Based Smart Environment System for Wireless Surveillance and Intrusion Detection (ABSESWSID) is implemented as a means of keeping 24 hours surveillance of the environment. An ABSESWSID is a wireless surveillance system that uses artificial intelligence and data communication tools to monitor and report activities in the remote environment. It is the responsibility of the artificial intelligence agents to sense the environment and deliver the information to other neighboring agents in the environment using data communication tools like wireless transceivers.

The work however was not tested using a real life hardware but a simulation of the environment. The implementation of the environment simulator was carried out using Object Oriented Analysis and Design Methodology tools. The implementation was done using C#.

1.1 Statement of the Problem

Pipeline vandalism has been a major issue facing oil production in Niger Delta Region of Nigeria. By estimation, in 2016 over 700,000 barrels of oil was lost on daily bases due to pipeline vandalism. Several methods has been proposed to combat these problems. The most popular of them has been to use community vigilante and law enforcement agents to guard the area. However, none of these yielded any lasting solution due to the disparity between the time an event occurs and the time a report is made for an action to be taken. If the vandalism in oil pipeline is to be contained then there is need for a real time surveillance system that will provide prompt information to the law enforcement agents and that is what this study Agent Based Smart Environment System for Wireless Surveillance and Intrusion Detection System (ABSESWSID) addresses.

1.2. Aim and Objectives of the Study

Our aim for embarking on this research work is to develop an agent based smart environment system for wireless surveillance and intrusion detection and the objectives are as follows:

- Provide a 24 hours, real-time surveillance of the remote environment
- Provide an optimized network performance
- Reduced energy utilization of the network nodes
- Provide network management interface to aid decision making
- Provide a network that will work without any guide.

1.3. Significance of the Study

Wireless Sensor Network (WSN), which, is a subset of this work, has been deployed in many application domains to keep surveillance and detection of intrusions. However, the system (WSN) has always been associated with one problem or the other, one of the most predominant is the deployment cost and power consumption. If the cost of the nodes are so high then it affects the feasibility of the project. The power consumption on the other hand affects the lifetime of the network. Such that if the area under consideration is very hostile for humans, either to visit for replacement of the sensor nodes or recharging of the nodes, then the aim will be defeated as the network will not be able to keep surveillance of events when all the nodes have exhausted their power supply.

These challenges have always made the deployment of WSN very inappropriate for unsafe environments. Thus, the significant of this work is that it is implemented to reduce power utilization and hardware requirement through intelligent routing decisions at each of the nodes.

1.4. Scope of the Study

This project is going to be implemented using C# and Microsoft SQL Server. Due to the difficulty in procuring the necessary hardware for this work, the testing was carried out in a simulated environment; in the simulation, the system provided the following features, which will be part of the real system.

- The system provides an interface through which the user can monitor all the activities happening at the remote station.
- The features provided by the system include remote monitoring of the energy level at the nodes, the power consumption rate, intrusion activities at the oil field and automatic redeployment of nodes.
- The implementation is based on Open System Interconnectivity (OSI) reference model (i.e. Application, Transport, Network, Data Link and Physical Layers).
- The nodes work unattended, taking autonomous decision in reporting events and moving close to signal when there is a hole, (i.e. having some areas that are not covered because of dead nodes) in the network.

2. LITERATURE REVIEW

Since the introduction of intelligent agent by McCarthy between 1956 and 1958 the field has attracted several researchers. Ever since then many authors have given the definition of intelligent agent but until date, none has been credited to have given a universally acceptable definition of the term intelligent agent. Though no definition has been universally accepted, many still believe that autonomy is an attributes that is central to all agents. This is shown in the work of Maes (1995) where she defines autonomous agent as “the computational systems that inhabit some complex, dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed” (Maes, 1995). The environment presents a platform from which the agent senses and acts. Without the environment, the agent will not have a place to execute its actions, and from the definition of Maes (1995) it is obvious that the environment creates a podium for the agent to sense and act. Furthermore, to support the importance of autonomy in agent computing

Mark and Gordana (2009) mentioned autonomy as one of the important concepts in the development of a variety of fields in computing (Burgin & Dodig-Crnkovic, 2009). The definition given by Brustoloni (1991)also centered on autonomy (Brustoloni, 1991). In his definition, he defined autonomous agents as “systems capable of autonomous, purposeful action in the real world.” In the definition given in an online white paper by Virdhagriswaran of Crystaliz, Inc. the term agent was defined to represent two orthogonal concepts “The first is the agent's ability for autonomous execution. The second is the agent's ability to perform domain oriented reasoning” in this definition also autonomy is still the focal point. Aaron and Lakshmi say that one of the purposes of an agent is to satisfy its user by autonomously and continuously performing his (the user's) task. Autonomy is central here because the agent has to satisfy the users request without the user being present (Hector & Narasimhan, 2005).

In the definition given by Wooldridg, he states that an agent is “a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.” (Wooldridg, 2002) Like we mentioned earlier the agent needs an environment to receive input and perform some actions. Just like Wooldridge, Russell and Norvig also gave a definition of an agent to include how the agent perceives and acts on the environment, they defined an agent to be “anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators” (Stuart J. Russell, Peter Norvig, 2010), this definition has been adopted by many authors though (Stan Franklin, Art Graesser, 1996) criticized it saying that since this definition depends on environment, sensing and acting it will make every program to be an agent. Since the environment acts as a means through which the program receives input and sends output. The means of receiving input can be called sensing while sending output can be called acting.

As we have seen many authors have incorporated environment to the definition of agent, now let's look at how the agent relate to the environment. The agent architecture is the foundation through which all agents are built. It helps in the modeling of the agent's behavior. Every reasoning of the agent is dependent on the architecture as this forms the building block of the agent's intelligence. The agent architecture uses the properties of the agent to enable it make a decisions. According to Wooldridg (2002), it is a software architecture with the intension of supporting the agent's decision making through the properties; reactive, proactive and autonomy (Wooldridg, 2002). Maes (1995) said that it shows the interaction between the sets of components which decomposition is supported by the techniques and algorithms included in the agent architecture (Maes, 1991). The architecture makes the agent to conceptualize the environment before it acts based on the information it has on its data structure or it may respond immediately to what is happening in the environment based on its impulse. The action the agent takes depends on the agent's architecture.

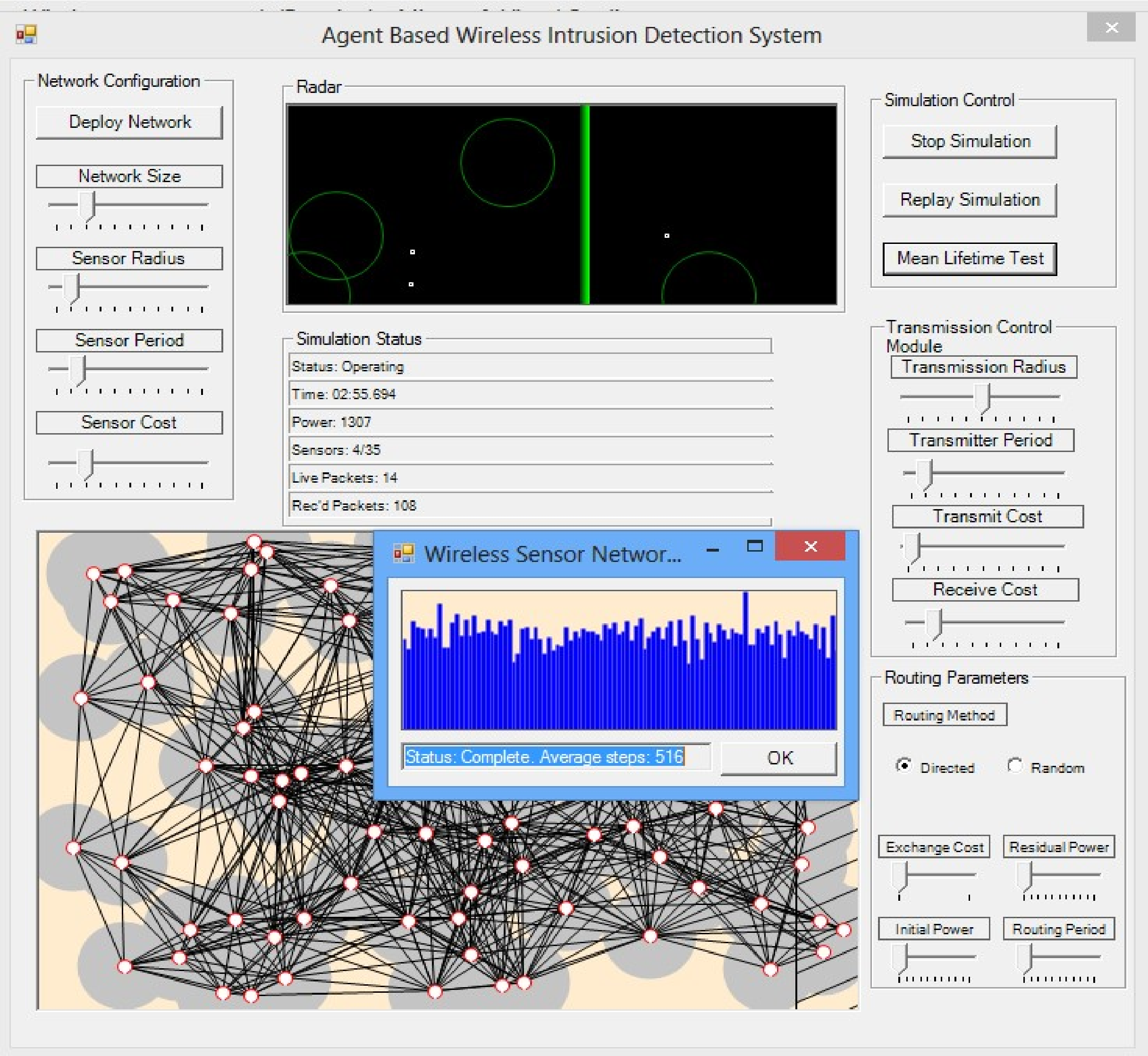
Over the years research has been going on in the area of agent architecture. The first agent architecture was symbolic or logic based architecture, this architecture has been around since 1956 though it has some draw backs which led to the discovery of another type of architecture called reactive architecture, this architecture has been around since 1985. Another architecture that came around 1990 was hybrid architecture, it combines the advantages of logic based architecture and reactive architecture. Another architecture developed by Bratman in 1987 is the BDI architecture. Broadly speaking the four architectures belong to the classical architecture. Other categories are the cognitive and semantic agent architecture. The BDI architecture is a deliberative agent architecture that has its basics on the three state mental characteristics; the belief, desire and intention.

3. SYSTEM IMPLEMENTATION AND TESTING

In this section, we will be testing the application to verify that the requirements outlined in the use case are met. In our test, we will be looking at the test plan from the point of view of the various modules that make up the application. Before we proceed, we will discuss the application-testing interface and the simulation purpose.

i. Application Testing Interface

The real life testing of the system is going to involve hardware, which is going to be very costly, so we are not going to embark on it due to the unbearable cost of the hardware. Therefore, we are just going to simulate the test environment. The simulation environment is build using visual studio 2010. The interface is as show in figure.1. It comprises of four main modules the network configuration module, the simulation module, the transmission control module and the routing parameter module. Apart from the modules, the system consists of the display, which has the network, the radar where all that is happening at the display is viewed life. Included also is the simulation status that reports all the activities that is happening at the display as it is running. The main purpose of the simulation is as follows:



Simulation Status

Status: Operating

Time: 02:55.694

Power: 1307

Sensors: 4/35

Live Packets: 14

Rec'd Packets: 108

Status: Complete. Average steps: 516

● Directed ○ Random

Figure 1: The GUI of the Agent Based Wireless Intrusion Detection System

ii. Simulation purpose

This application is a simulation of the wireless intrusion detection system. The network may be deployed based on a wide range of parameters: network size (number of nodes), communications distance, energy costs for transmitting and receiving packets, etc. The network can then be used to simulate the detection of vectors traveling across the sensor network field. In this simulation, when a vector trips the sensor of a network node, the node generates a data packet and sends it to a downstream network node. The packets are routed appropriately until they reach a sensor within the "uplink zone" (the right side of the map, designated with a striped pattern.) Each node also simulates an energy store, which is depleted by sending receiving packets, and by detecting vectors. Since the nodes have finite energy, they will eventually power down and drop out of the communications network, causing network failure.

The idea behind this network is that it can be deployed simply by scattering sensor units across the area, e.g. by dropping them out of an airplane; the sensors should automatically activate, self-configure as a wireless network with a mesh topology, and determine how to send communications packets toward a data collector (e.g., a satellite uplink.) Thus, one important feature of such a network is that collected data packets are always traveling toward the data collector and the network can therefore be modeled as a directed graph (and every two connected nodes can be identified as “upstream” and “downstream.”). A primary challenge of such a network is that all of the sensors operate on a finite energy supply, in the form of a battery. (These batteries can be rechargeable, e.g. by embedded solar panels, but the sensors still have a finite maximum power store.) Any node that loses power drops out of the communications network, and may end up partitioning the network (severing the communications link from upstream sensors toward the data collector.) Thus, the maximum useful lifetime of the network, at worst case, is the minimum lifetime of any sensor.

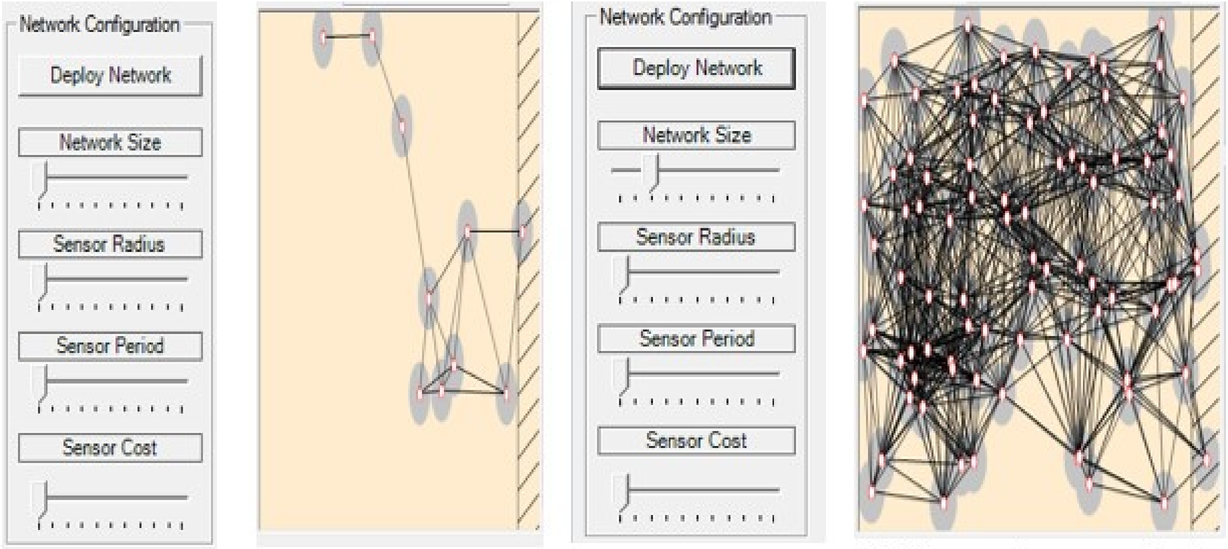
3.1. Testing Plan

The functionality provided by the application is grouped into modules: the network configuration module, the simulation control module, the transmission control module and the routing parameters. In most cases, the modules interact in order to carry out their designated assignments. Testing of the modules will form our test plan.

Network Configuration Module

This module determines the hardware properties of the network. The following variables can be configured:

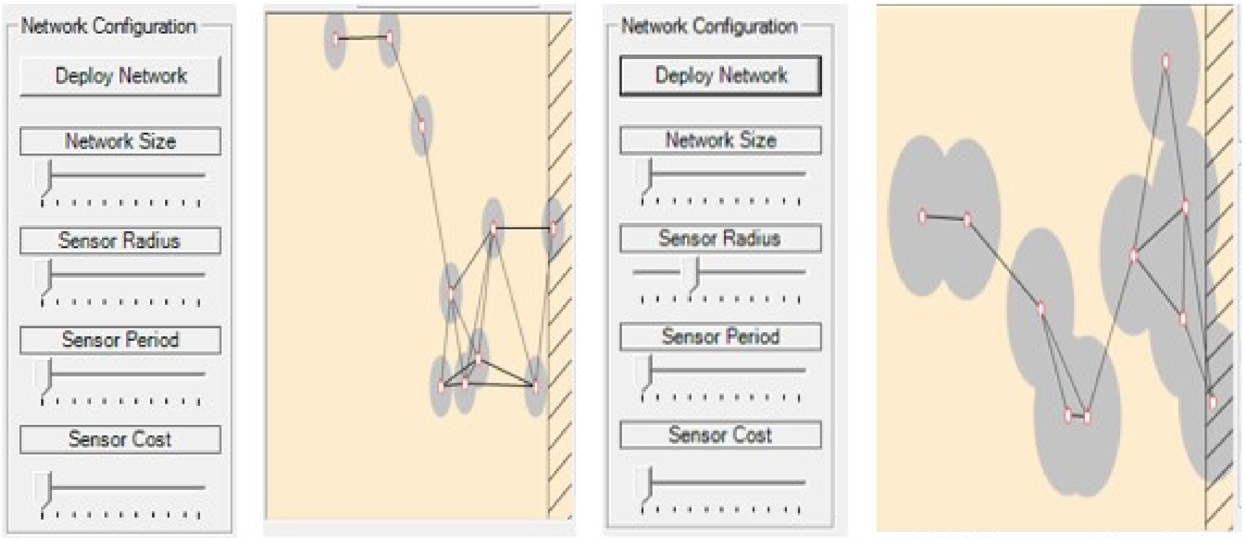
Network Size: The number of nodes in the network. If set to a high value, the network will have several hundred of nodes; and since this will hugely increase the density of the network and the number of network connections, this may bog down the simulation. If a large network is desired, it is recommended to reduce the Transmission Radius. See figure 2 (c) for the setting and (d) for the output.



(a) Initial Setup (b) Initial Output (c) Network size adjustment (d) Output after network size adjustment

Figure 2: Setting up of the network size and deployment

Sensor Radius: The proximity range of the sensors in the network. When set to high it increases the reception range of the sensor nodes. See figure 3.



(a) Initial Setup

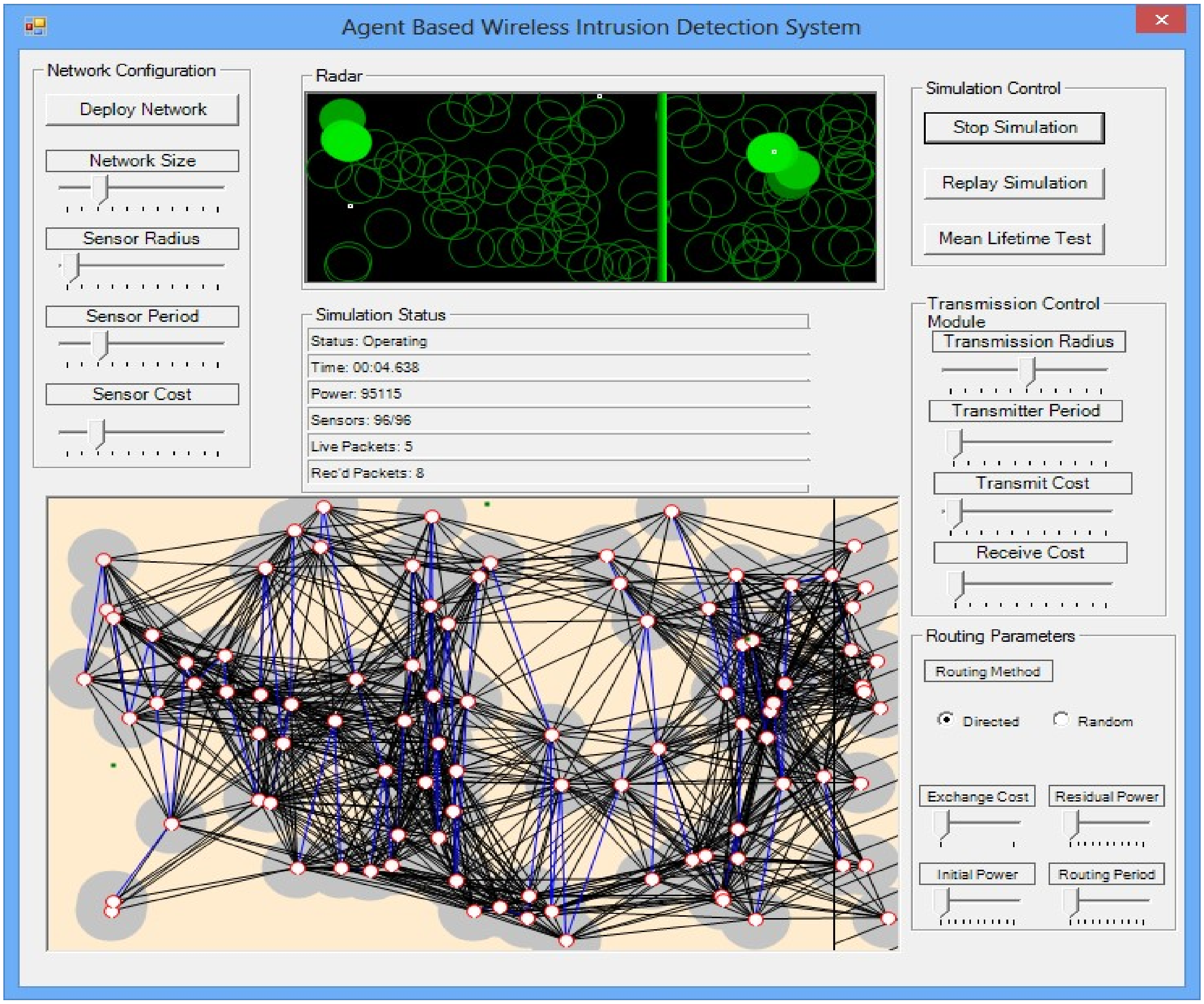
(b) Initial Output

(c) Sensor radius setup

(d) Output after the deployment

Figure 3: The setup and output of the sensor radius

Sensor Period: The delay period between sensor detecting events. If set to a low value, a network sensor will fire rapidly as a vector enters its sensor radius (thereby consuming a lot of energy.) If set to a high value, the network sensor will wait a long time between firing a second packet, see figure 3.



Simulation Status

Status: Operating

Time: 00:04.638

Power: 95115

Sensors: 96/96

Live Packets: 5

Rec'd Packets: 8

● Directed ○ Random

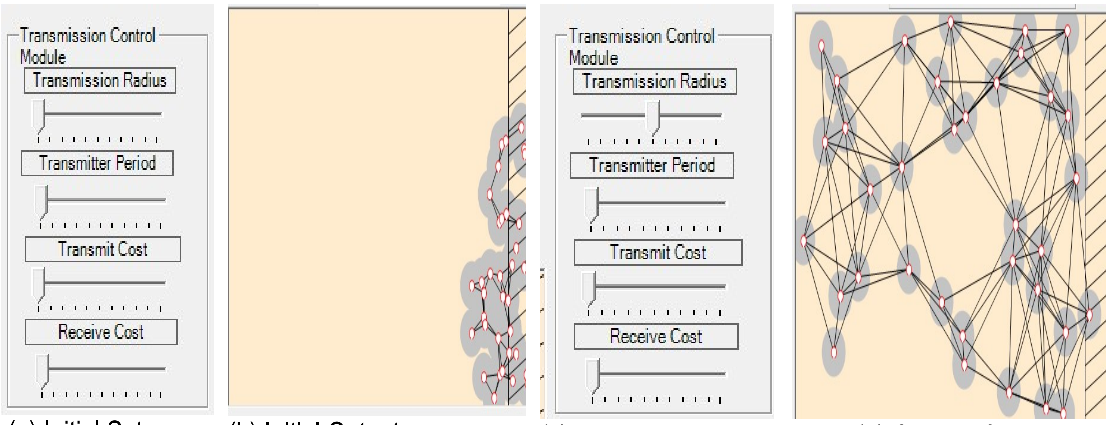
Figure 4: Sensor Period and Cost setup

Sensor Cost: This shows the energy cost in detecting a vector and generating a packet. As a means of demonstration, see figure 4.4.

Transmission Control Module

This module determines how the packets are being transmitted on the network. In other to effectively control the transmission of data across the network, the following variables can be configured:

Transmission Radius: The maximum distance within which two network nodes can communicate. If set to a high value, nodes on opposite sides of the map may be able to reach each other; if set to a low value, nodes must be very close to communicate, see figure 5.



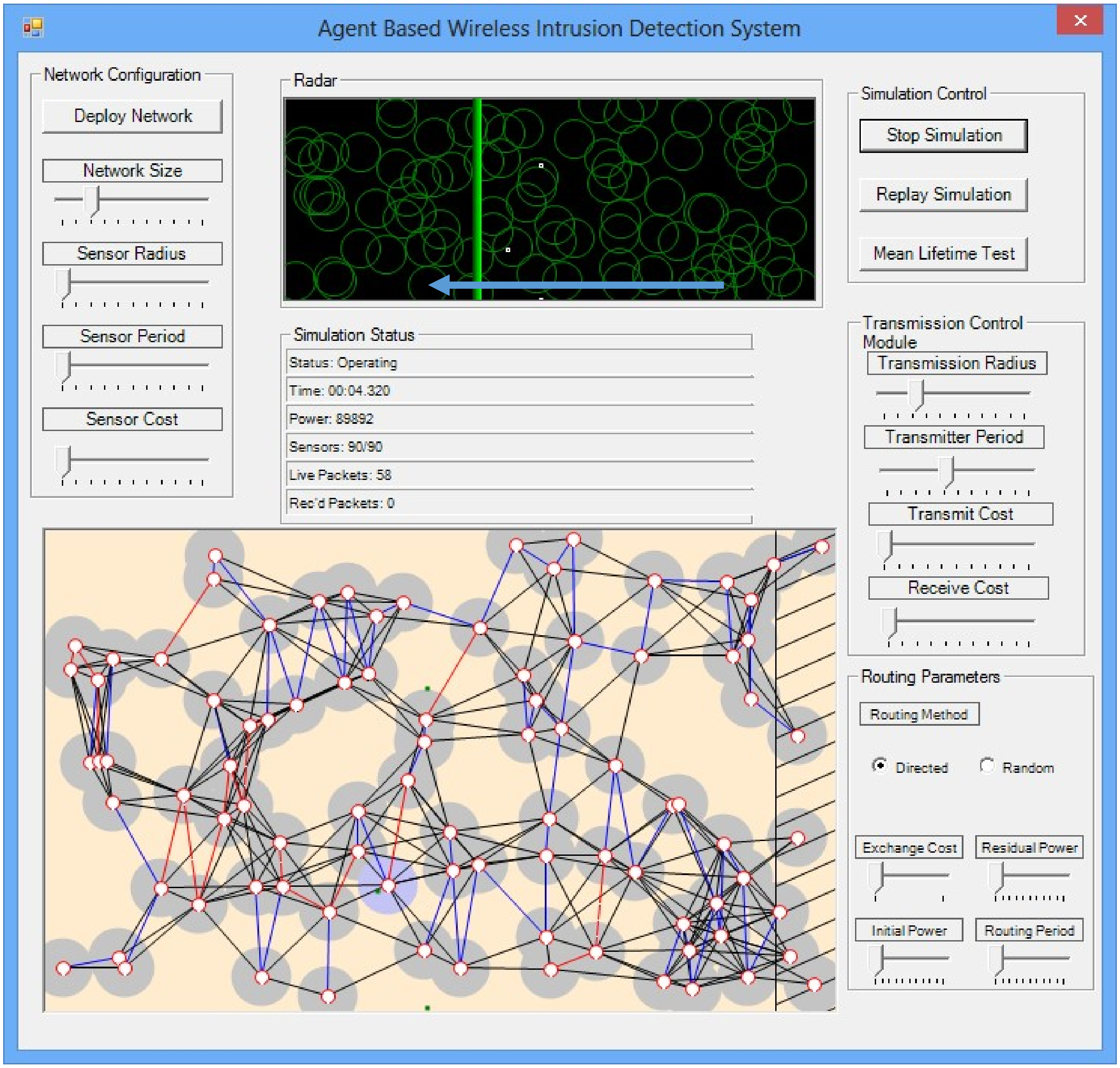
(a) Initial Setup (b) Initial Output (c)Transmission radius setup (d) Output after the deployment

Figure 5: Setup of the transmission radius in the transmission control module

Transmitter Period: The amount of time required to send a packet. Setting this to a high value will cause each packet transmission to take several seconds. Thus, the data received at the radar will be quite stale, since many seconds will have elapsed since the triggering event. However, the high period allows the user to monitor the packet-exchange process on the network map, see the simulation status in figure 6.

Transmit Cost: The energy cost in sending a packet. Setting this value very high will cause nodes to be depleted after sending only a few packets; setting this value very low allows the nodes to send many hundred packets. (Note that this is always scaled based on the distance between the nodes; thus, since more distant nodes can only be reached by a more powerful broadcast, such transmissions more quickly deplete the energy store of the transmitting node.), see the simulation status in figure 6.

Receive Cost: The energy cost in receiving a packet. (This value is not scaled, as is the transmit cost.)



Status: Operating
Time: 00:04.320
Power: 89892
Sensors: 90/90
Live Packets: 58
Rec'd Packets: 0

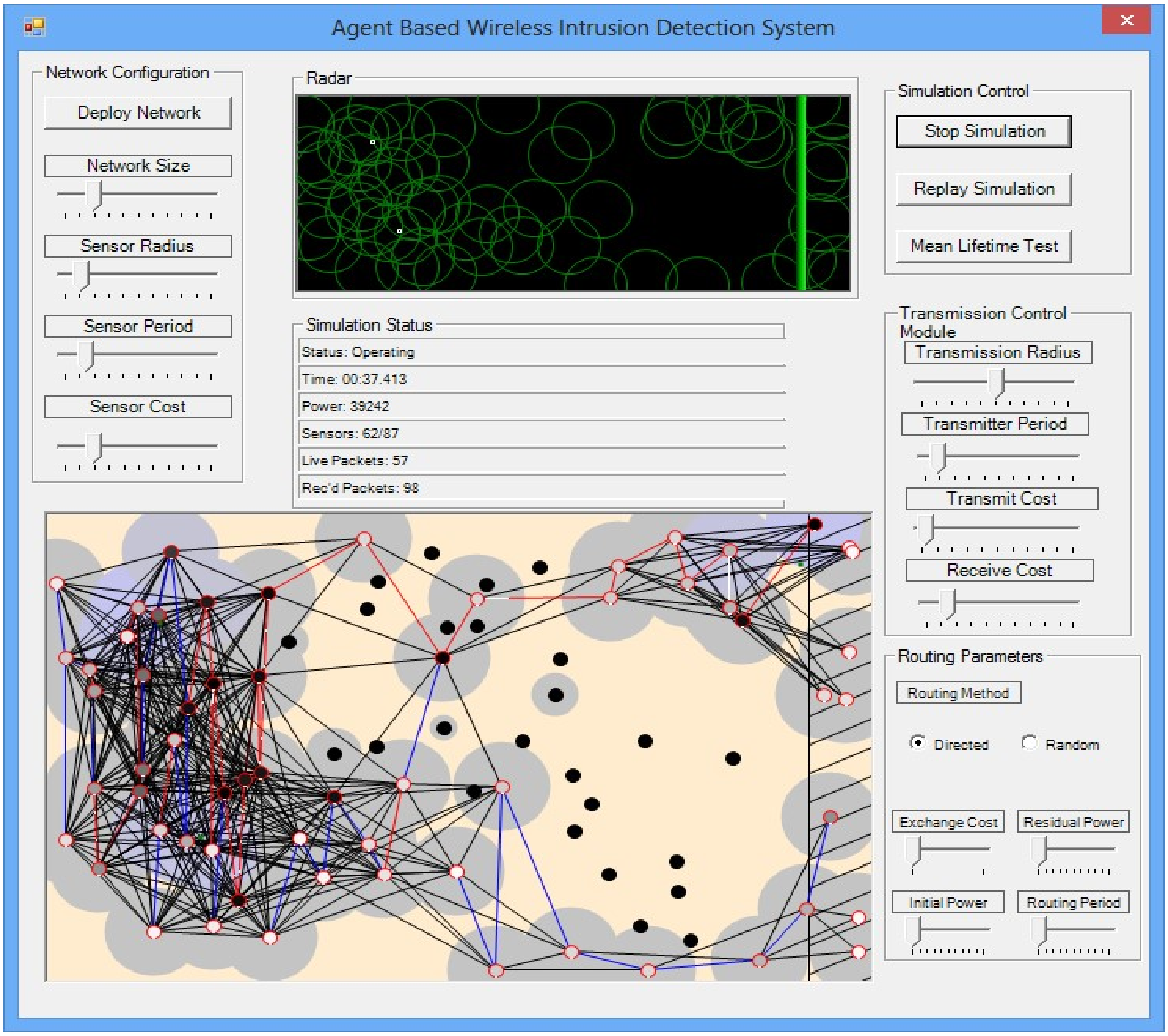
● Directed ○ Random

Figure 6: An application interface showing the simulation status of the network

Simulation Control Module

This module consists of three control buttons: the start simulation, replay simulation and mean network lifetime.

- **Start Simulation:** Once the network has been deployed, the simulation may be run by clicking “Start Simulation.” The map will show vectors moving through the field and triggering sensors. The sensors may run out of power and drop out of the network, and eventually, all nodes will be powered down. The progress of the network can be monitored via the “Simulation Status” box. A new simulation may be run by stopping and restarting the simulation. Alternatively, the previous simulation may be reviewed by clicking the “Replay Simulation” button, see figure 7.



Simulation Status

Status: Operating

Time: 00:37.413

Power: 39242

Sensors: 62/87

Live Packets: 57

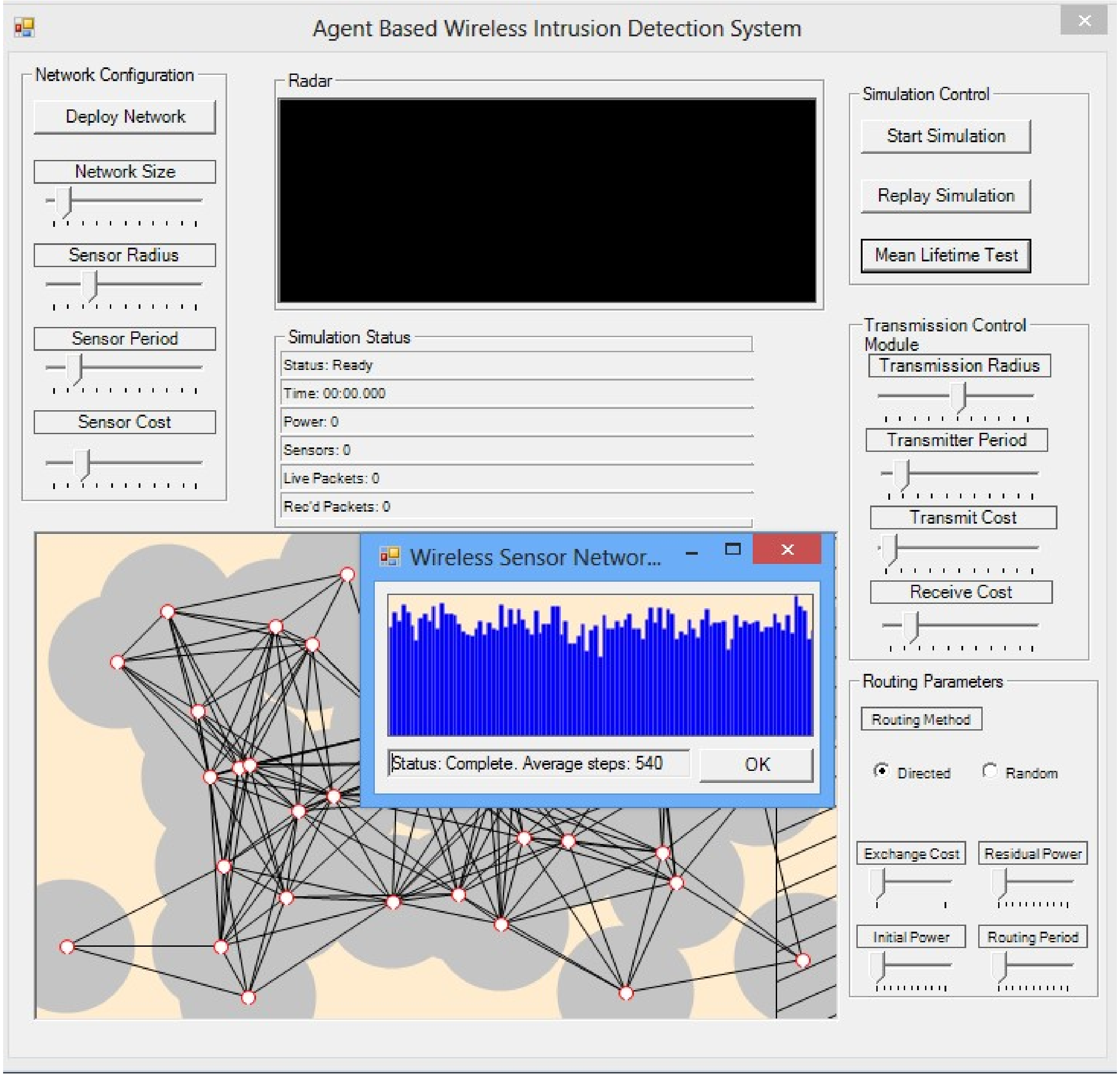
Rec'd Packets: 98

Routing Method: ● Directed ○ Random

Figure 7: Testing the simulation control module

Mean Lifetime Test

The application has the ability to run successive tests on a network and report the mean network lifetime across 1,000 trials, see figure 8.



Status: Ready

Time: 00:00.000

Power: 0

Sensors: 0

Live Packets: 0

Rec'd Packets: 0

● Directed ○ Random

Status: Complete. Average steps: 540

Figure 8: Mean Lifetime Test

Test Data

This project is a simulation of the agent based wireless intrusion detection system; the test data is obtained during the simulation of the network. The test data are the vectors, which represent the intruder in the network. The vectors are generated after the network has been deployed and simulation has started. When the simulation starts the activities of the test vectors generates some packets based on the detected vectors, the location of the node that detected the event, the date and time when this event occurred, etc. The network and vectors used as test data in the simulation is shown in figure 9.

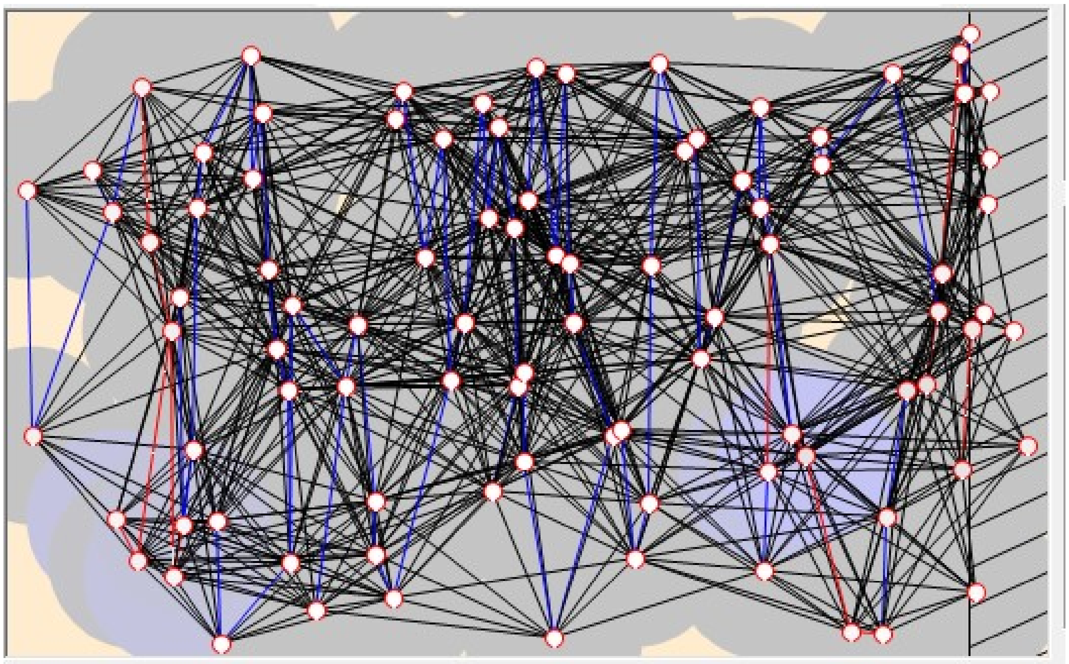


Figure 9: The network and vectors used as test data

.

4. SUMMARY

Agent Based Smart Environment for Wireless Surveillance and Intrusion detection System is an intrusion detection system that monitors and report all the activities happening at the remote environment (the oil field). This work looked at how to optimize the network performance by implementing the node in a way that it will make intelligent routing decision that will optimize the network performance. As a way of introduction a background, statement of the problem, scope and limitation of the work where discussed. A review of all the work related to this project has been analyzed, after which a detailed explanation of the methodology and material used for the project where presented. A simulation of the system was developed to test the desirability of the results.

5. CONCLUSION

This work addressed performance optimization and a considerable reduction in power utilization. Power and performance have been a major issue in Wireless Sensor Network, but this issue has been laid to rest as this work has created a model that helps the nodes to make some intelligent decision during packet routing so that performance can be enhanced. To reduce power utilization the nodes are made to respond to some events, which helps in preserving their energy reserve.

6. RECOMMENDATION

Agent Based Smart Environment for Wireless Intrusion and detection system is a very fertile research ground as performance and power continue to be a major factor affecting development in this area.

REFERENCES

1. Brustoloni, J. C. (1991). Autonomous Agents: Characterization and Requirements.
2. Burgin, M., & Dodig-Crnkovic, G. (2009). A Systematic Approach to Artificial Agents.
3. Hector, A., & Narasimhan, V. (2005). A New Classification Scheme for Software Agents. *Proceedings of the Third International Conference on Information Technology and Applications.* IEEE.
4. Maes, P. (1995). Artificial Life Meets Entertainment: Lifelike Autonomous Agents. *COMMUNICATIONS OF THE ACM*, 108-114.
5. Omotala, J. S. (2016). "Liberation movements" and rising voilence in the Niger Delta: The new contious site of oil and environmental politics. *Studies in Conflict and Terrorism*, 33, 36-54.
6. Stan Franklin, Art Graesser. (1996). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*.
7. Stuart J. Russell, Peter Norvig. (2010). *Artificial Intelligence A Modern Approach.* New Jersey: y Pearson Education.
8. Wooldridg, M. (2002). *An Introduction to MultiAgent Systems.* West Sussex: JOHN WILEY & SONS, LTD.