

# *Complex Adaptive WSNs for Polluted Environment Monitoring*

*Amjad RATTROUT*

Department of Computer Science  
Arab American University  
Jenin, Palestine  
[Amjad.ratttrout@aaup.edu](mailto:Amjad.ratttrout@aaup.edu)

*Hamzah Hijawi*

Department of Computer Science  
Arab American University  
Jenin, Palestine  
[hamzah.hijawi@gmail.com](mailto:hamzah.hijawi@gmail.com)

**Abstract**—The development of low cost, low power, multi-functional sensor nodes created a new form of networking applicable to a variety of fields. The goal is to create low cost, low power consumption and web accessed wireless sensor nodes in order to be used for environment monitoring. A wireless sensor network consists of large number of sensor nodes. One of these nodes is used as a gateway to be connected to the internet; this node is called a base station or a sink node. The capabilities of the gateway node will be larger than the other nodes in the system. However it will be directly connected to the internet via Wi-Fi or Ethernet adapter. Moreover a web server may be installed on this node; this will enable the system to be directly accessed any time from any location. The low power consumption is strongly needed to be used in A wireless sensor networks; this because the wireless sensor nodes almost powered by batteries. In order to achieve this goal an intelligent hardware and intelligent software should be implemented carefully to adapt the environment changes and do some adjustments and calibrations on the system.

**Keywords**—WSN, Multi Agents, Complexity, Co-evolution

## I. INTRODUCTION

Wireless sensor networks are emerging technologies of the past few years; they involve a large number of small nodes. Each node senses environmental changes and report them to other nodes over flexible network architecture. They are varied in their applications and areas [1] [2]. The main components of a wireless sensor node are microcontroller, transceiver, and sensors. Microcontroller processes data collected by the sensors, controls other components in the node and performing power consumption management. Transceiver transmits/ receives data from/ to other nodes in the network. Sensors have the sensing tasks. Wireless nodes are powered by batteries.

The main challenges are to produce low cost, low power consumption and web accessed nodes [3]; generally, the transceiver consumes the largest amount of power. Therefore, it is an advantageous to send data only when it is required. This requires intelligent agents loaded to the sensor nodes which can convolute and make decisions according to the system status. Additionally, it is important to minimize the power consumed by the sensor itself. Therefore, the hardware should be designed to allow the microcontroller to judiciously control

power consumed by the transceiver and controlling the sensing frequency; however if the sensed phenomena is not critical, the microcontroller must be programmed to read the sensors only when a reasonable changes occur in the environment [4].

Wireless sensors networks are autonomous systems with severe energy and processing power limitation and constrain, however end nodes have limited reliability. In such conditions, self-organizing, energy efficient, fault tolerant and adaptive algorithms are required to be used in WSN [1].

Generally the main tasks of a wireless sensor node are to interact with it is environment (sensing) and reporting sensed data to the sink node. The sensing capabilities of different environment phenomenon emerge new applications for WSN; this includes: Environmental monitoring, health care systems, military, educations and smart home buildings [5]. WSN simplify the systems managements and helps building a smart world.

WSN systems can be deployed with different topologies; the most common are Bus, Tree, Ring and Star. In Bus topology all nodes connected via single link and when a node wants to communicate with another node, it sends a broadcast

message, all nodes connected to the bus will receive the message and only the destination node will process the message. In Tree topology the system is divided into levels, however nodes only communicate with their parents. In Ring topology every node has only two neighbors, in this topology the message is propagated to each neighbor until received by the destination node. In Star topology the sink or main node is logically located at the center point of the system; it will have a direct connection to each node within the system and nodes communicating by passing messages through the sink node [6]. Star topology will be used to build the clusters in our system.

In this research, we will focus in the software intelligence part and build smart agents to control the system. The agents will use some mathematical calculations to decide which nodes are redundant and putting them in the sleep mode. Nodes in the sleep mode will wake up periodically and check if they still redundant. The rest of this research is organized as follow. Related work is given in section 2. Section 3 gives an overview about the system architecture used. And the problem is defined and discussed in section 4. Section 5 discusses the proposed method to implement the system using the multi agents. In section 6 we propose two mechanisms to reduce the power consumption in the system. Section 7 is the simulation part and finally conclusions are discussed in section 8

## II. RELATED WORK

The problem of intelligent implementation of wireless sensor network had been studied extensively in that past few years [1] [3] [4]. Most of researches focus on the small part of the system [7] [8] and almost none of them provided a complete system solution. Wireless sensor networks threads were studied in normal client server architectures [2] and multi agents approaches were discussed in [4] [8] [9]. The complexity and adaptation in multi agents systems were extensively studied in Web systems [10]. However in [11] they discussed the complexity in data collection, data aggregation and data selection over the tree model in which the sink node which is the first level in the tree is connected with two nodes to form the second level and so on. The conative WSN was studied in [12], however in this model the network can make decisions and actions based on the condition of the environment and the current system status to achieve its goals. BDI model which is used to implement the intelligent agents was introduced in [14]; they described the general architecture of this model and defined the belief, desire and intention in the wireless sensor node, using this model the agents will be aware of their environment and can cooperate together to achieve the system goals. An example was introduced in [16] in which the wireless sensor network was used to implement an intelligent transportation system.

In this research we will provide a complete solution for implementation intelligent wireless sensor network using multi agent approach and will show the system components, types of agents and their interaction.

## III. SYSTEM ARCHETICTURE

The system will be built based on a cluster topology in which for each cluster there is a cluster head node which is responsible for coordination and data aggregation from other member nodes. Cluster head nodes have direct connection to this sink node. Sink node considered as a gate way of the system. However it is the connection between our environment and the external word. Sink node and cluster head nodes have more processing power and longer life time from other nodes in the system because they are considered as key nodes and have more extra work to do. Sink node is connected to the external word using internet and the collected data will be stored in external data base.

For simplification we considered that the system has three clusters and nodes have direct connection to their cluster head.

Fig. 1. System architecture

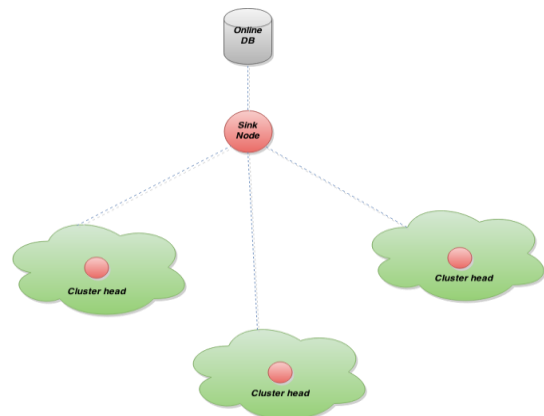
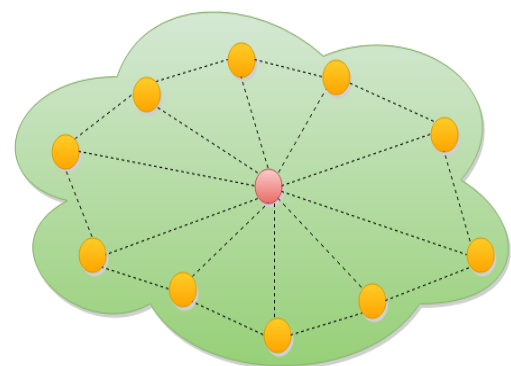


Fig. 2. Cluster architecture



## IV. PROBLEM DEFINITION

Wireless sensor network is an open complex adaptive system, in the following sections we will describe the most

important characteristics of WSN and we will propose a method using intelligent agents to resolve those issues.

#### A. Environment Description

The main application of WSN is environment monitoring, and in most cases it is the natural environment [2]. WSN gives humans clear observation and detailed studies about the surrounding environment by collecting data using sensors. Applications involved in the environment monitoring are developed from data we get from sensors to information after processing the data and finally to knowledge the human gains about the real status of his environment. Application scenario explained in this research is related to collecting data from a polluted environment, this type of applications requires collecting data much as possible. The collected data is processed to give clear information or can acts as input to another system in order to do some actions on the environment. For simplification measuring CO<sub>2</sub> concentration will be used as an example, however any other example is applicable.

#### B. Complexity of WSN

Wireless sensor network is a complex system; complexity can be seen in many fields. However Data collection, data aggregation, processing the gathered information, time complexity, messages complexity and energy cost complexity are the most complex elements in WSN [10]. Different types of components and large number of nodes make the complexity in systems like WSN. However in data collection, we need to decide which data we need to collect from sensor nodes and how to aggregate the raw data, this includes all kinds of queries and routing algorithms used to deliver data to the base station node. Processing data at sensor nodes requires complex operations; however do we need to process all the data or just pick up some samples. Reduction energy consumptions in sensor nodes almost the most important feature in WSN, however many algorithms are used to decide when and how to stop sensing, increase/ decrease sampling rate, even sleep the node entirely. To make the system simpler and function well, a kind of decisions making needed to be implemented as will be shown in next sections.

#### C. Adaptation and Co-evolution

WSN always interacts with its environment; however it should adapt to the environment changes and convolutes to increase the system life time [10]. Kinds of adaptations include decrease the sampling rate when the recent sensed data has the same values. Another kind of adaption in this system is the election processes to elect a new cluster head node when the old node wants to die due to low remaining battery power. The aim is to make an intelligent system which able to make decisions and actions based on the conditions of the environment. Intelligence when applied to WSN can make the system behaves better and will increase the performance [12]. The system should be aware of the sensed data, when and how to forward it, this will result in better power and bandwidth management over the overall network. WSN implementation using agents will enable the system to have a high level of knowledge about the environment and the type of information being exchanged, this will help in achieving the system

objectives by making the network aware of and adapts to the application requirements and the environment in which it is deployed.

### V. MULTI AGENT APPROACH IN WSN

Traditional methods such client-server architecture used in WSN in which each node senses the environment independently from other nodes and propagate the data to it is neighbors until reaching the sink node can work with small size networks, however this approach has many issues; what if the system get larger and larger and we need to cover more areas, all nodes are active and they flood the sensed data in the network, this will put a heavy load on the other nodes because they have additional task to propagate other nodes data and as a result of that network performance will decrease. Even more, what about the sink node capability to handle and process this huge amount of data coming at the same time? Bottleneck will appear at some points causes unexpected faults [4]. Using multi agents in the sensor networks can solve the scalability, transparency and performance issues discussed previously. Redundancy elimination and saving power consumption also can be achieved with a good implementation of intelligent agents. However for each specific task in the system a specific type of agents will be created [8].

Belief, desire and intention (BDI) model is the most popular model for implementing intelligent agents. Multi agents may exist in a single complex system and almost they have common task which performed by each of agents independently from others. Agent may communicate with each other via messaging protocols [13]. However the goal is to increase the level of abstraction by using belief, desire and intentions instead of instruct the agent exactly what to do and how to do. Intelligent agent should be able to decide and make decisions according to the system variables.

#### A. Agents Types and Functions

Taking in consideration that following tags are used:

- S\_NODE: Tag to identify the sink node.
- H<sub>i</sub>\_NODE: Tag to identify head cluster node, where i is the cluster number.
- N<sub>ij</sub>\_NODE: Tag to identify each node within its cluster, where j is the node number.

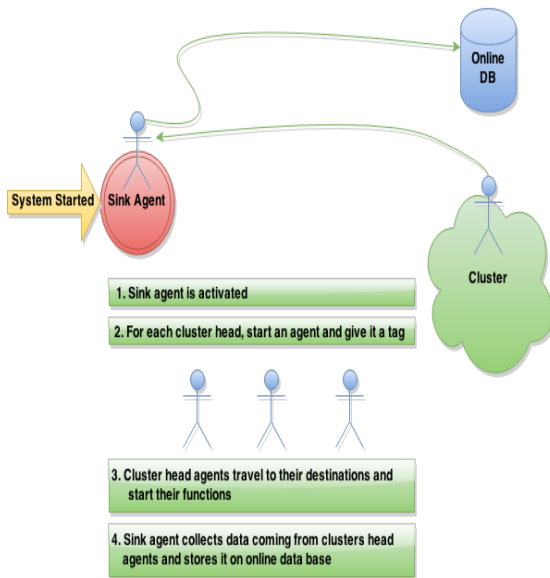
System is divided into multiple agents, each agent is specialized only in one function and he will die after complete his task. Following are the intelligent agents used in our system:

- Sink agent (Main system agent).
- Head clusters agents.
- Environment sensing agents.
- Control agents.
- Tagging agents.
- Data aggregation agents.

*Sink Agent (Main Cluster Agent)*

When the system runs for the first time, the main agent will be activated. Simply main agent will be the brain of the system. It will initiate tagging and main cluster agents, in addition to collect the aggregated data and stores it on online data base.

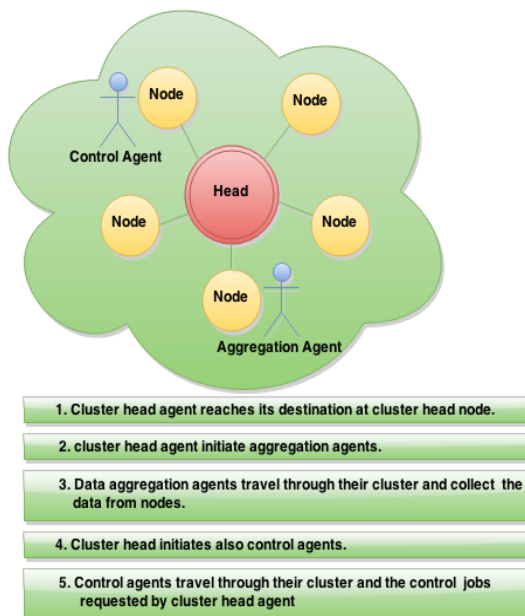
Fig. 3. Sink agent flow chart



*Head Clusters Agents*

Head cluster agents are created at sink node and travel to their destinations at cluster head nodes. All other agents in the cluster are created in the cluster head and they travel to sensor nodes in order to collect data or do some controls within their cluster.

Fig. 4. Head cluster agent flow chart



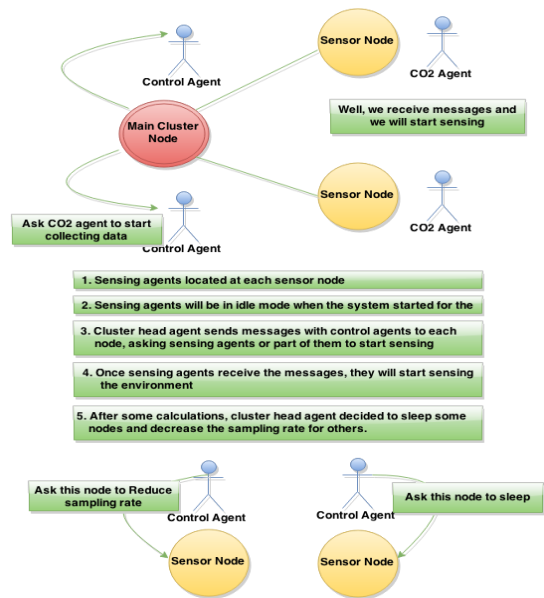
*Sensing Agents*

Sensing agents lie on each sensor node and they have only one function which is to read an environment variable, however for each variable there is a special agent, this includes agents responsible for reading CO<sub>2</sub> concentrations, agents to read VOCs, etc. Sensing agents started their work upon request from the head cluster agent through the control agents.

*Control Agents*

Control agents are initiated from head cluster agents, their job is to control the cluster nodes as requested from head cluster agent, this includes checking redundant nodes and put them in the sleep mode and stop/start sensing some environment variables.

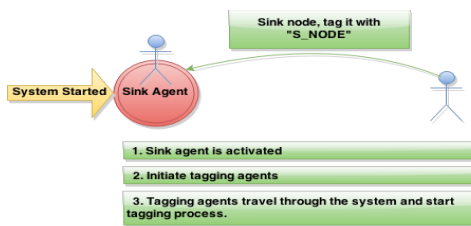
Fig. 5. Control and sink agents flow chart



*Tagging Agents*

Tagging agents are initiated at sink node using the sink agent and also at the cluster head nodes using cluster head agents. However tagging agents created at sink node are responsible for tagging the sink node and the cluster head nodes. Whereas tagging agents created at clusters head are responsible for tagging each node in their cluster. Tagging agents have additional task which is to increase the revision number of each sensor node visited by aggregate agents during data aggregation process.

Fig. 6. Tagging agents flow chart



Data aggregation agents

Data aggregation agents travel through the network and collect data from sensor nodes. However each sensor node has small database. When the aggregation agent reaches a sensor node, it will concatenate that table to its current table and empty the local table.

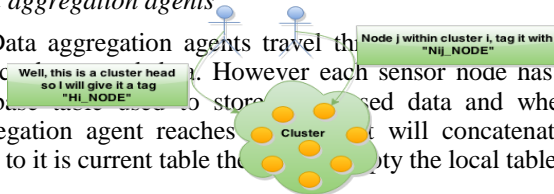
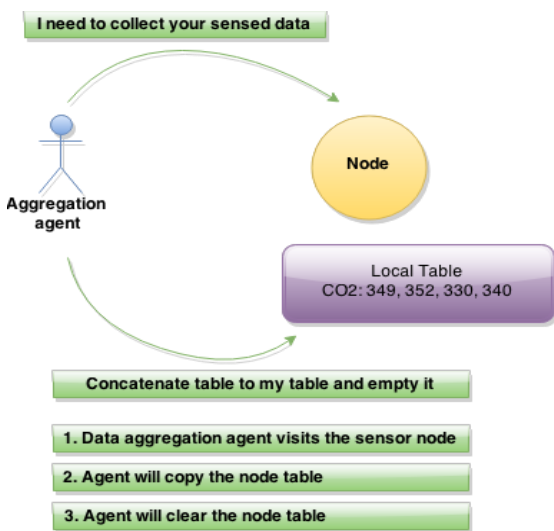


Fig. 7. Data aggregation agents flow chart



B. Agents Interaction

Most applications in WSN use multiple mobile agents to collect, share and process information about large scale environment. Information collected by agents need to be reported to the external users through the system gateway which is the sink node. Collecting information by agents and send it back to the sink node only when it is necessary will take less power than each agent sends his information alone [8] [9]. So how the system internally works? How the agents interact and cooperate to achieve the system tasks? First, when the main agent start running, it will have information about the number of clusters in the network and the clusters head IDs, actually only this information is needed for the main agent to start his work. So it will create one cluster head agent for each cluster. Cluster head agents travel through the network until each of them reaches its destination, however the agent knows his destination by comparing the cluster head ID he carries

with the node ID, if it is the same then the agent will be activated on the main cluster node and start managing his cluster.

Cluster head agent only needs the information about what environment variables should be collected by the sensor nodes, so for each variable it will create sensing agent which will travel to the sensor nodes. Sensing agents will start sensing tasks immediately when they reach the nodes, however each agent has small table to store the sequence of collected data. At some point the cluster head agent decided to collect the sensed data, so he will create multiple aggregation agents which will travel through the nodes. When the aggregation agent reaches the sensor node it will append the node tables to its table. Aggregate agent will continue with this process until he reaches his maximum payload size, at this point he will return back to the cluster head agent which will process the collected data. Aggregation agent will die after this step.

But how aggregation agent knows if another agent already visited this sensor node and collected the data? The proposed idea is to give the aggregation agent revision number when the cluster head agent create them, when those agents visit the nodes for the first time they will ask the tagging agent to tag the node with same revision number. The revision number is an incremented value, so for the next time the cluster head agent wants to collect the data; he will create a new aggregation agents with incremented revision number. When the aggregation agent reaches the node he will compare his revision number with sensor node revision and will copy the node tables only when it is revision number is greater than the node revision, and then it will increment the node revision number.

Another type of agents created by cluster head agent is the control agents, we need this agent in order to control, manage and detect faults in the cluster. Suppose for some reasons – will be described in later sections – main agent decided to put some nodes in the sleep mode, so it will create a control agent which will travel to those nodes and asks them to be in the sleep mode for a specific period of time, moreover control agents may ask some sensing agents to stop collecting data.

C. Building Trust between Agents

Building a trust in multi agent system is a very important step that needs to be taken in consideration. The needs of trust appears in the complex adaptive systems such as WSN because of the openness and diversity of the system components, however agents may frequently enter and leave the system which results in changing of the system structure. In such conditions it will be very difficult for the agents to build a confidence communication between each other. It is worth to mention that in most multi agent systems it is impossible to build 100% trust between agents, trust level is relative to the system goals. However an example, when the aggregate agent visits the sensor node to aggregate the data stored in its local database, sensor node needs to trust the aggregate agent and be sure it is part of the system. To build such a trust first the cluster head agent should encapsulate the aggregate agent and add the cluster ID to its header, when the aggregate agent

reaches the sensor node, sensing agent will compare its cluster ID with the cluster ID provided from the aggregate agent and will provide its table only if they have the same cluster ID. Another model which can be used is to use a trust agent which acts as a broker; however when the aggregate agent tries to communicate with the sensing agent in order to collect its data it will be checked by the trust agent which will decide if it is trusted or not according to its tag and policies. However these are just examples to illustrate the trust between aggregate and sensing agents, many other models can be used [18].

## VI. POWER SAVING MECHANISMS

There are many methods and mechanisms to reduce power consumption in sensor networks systems. Intelligent hardware design and implementation can almost achieve a good percentage of this goal. However intelligent agents can also contribute to achieve this goal. However using BDI model, the following methods are proposed to decrease the power consumption in WSNs.

### A. Put Redundant Nodes in Sleep Mode

In order to estimate the value provided by a sensor node many mathematical expressions can be used to calculate the duplication in the system. In our system of monitoring polluted environment an intelligent agent will be implemented in the cluster head node, it will check if there is a redundancy in the cluster and put the redundant node in the sleep mode [3].

How it works? Taking in consideration that our system is built based on cluster topology in which for each cluster there will be a head node and there will be a direct connection between the cluster head and each of the other nodes on that cluster. Using the variance for calculations estimations:

- A cluster head agent will be assigned to each cluster.
- It will communicate with other agents in its cluster and ask them to calculate for example CO<sub>2</sub> concentration at that point.
- Each node will send its calculated value back to the cluster head agent.
- The cluster head agent will save temporary the CO<sub>2</sub> concentration with node ID for each sensor node.
- The cluster head agent will calculate the average of CO<sub>2</sub> concentration based on its value and the other nodes values.
- If the difference between the average value and the real value is within the error range, then put that node in a sleep mode.
- Nodes in a sleep mode will wake up periodically and check again if they still redundant

### B. Reduce Sampling Rate

Sampling rate in wireless sensor network depends on the application, if we are dealing with critical environment and we

are concern about the recent data; high sampling rate should be used. However low sampling rate can be used for less critical applications such as environment monitoring which in most cases don't do any action on the environment. In this research which studies the polluted environment we use the BDI model to decrease the sampling rate because in such applications the data doesn't change often, so there is no need to have high sampling rate which consumes battery power [17].

Reduction of sampling rate is the control agent task which is described in the following steps:

- Cluster head agent who has the most recent collected data sees that the data values almost the same so he decides to reduce the sampling rate.
- Cluster head agent creates multiple control agents.
- Control agents travel through their cluster and ask the sensing agents to reduce the sampling rate.
- Control agents will die after finishing their task.
- Same procedure if the cluster head agent decided to increase the sample rate.

### C. Ask Sensor to Stop Sensing the Environment

Based on our model in which each wireless node is responsible for sensing more than one variable. Sometimes one of the sensors reports duplicated data or we are concern about the sensed data over a specific period of time. In this cases control agent interacts with sensing agent and asks it to stop sensing.

- Sink agent receives a request from network admin to stop sensing CO<sub>2</sub> concentration.
- Sink agent creates multiple control agents loaded with a message to stop sensing CO<sub>2</sub>.
- Control agents travel through cluster head nodes and deliver the message.
- Each cluster head agent creates multiple control agents which travel through the cluster.
- Once the control agent reaches the sensing node, it contacts with agent responsible for CO<sub>2</sub> sensing.
- Control agent will die and the sensing agent will stop sensing.

### D. Ask nodes to Stop Processing Collected Data

At the critical point, when the overall remaining power in the system is very low. One approach to increase the life time is by doing the calculations and processing in the external world, this simply happens by asking the sensing nodes to forward the raw data collected from the environment.

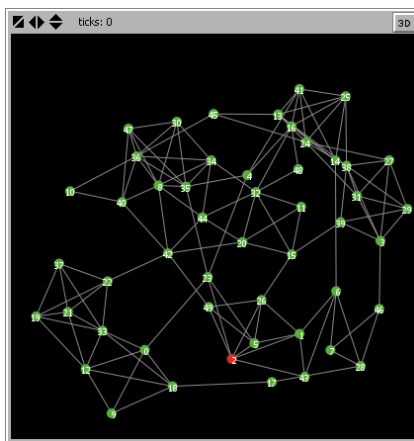
- Sink agent creates multiple control agents loaded with a message to stop processing data.
- Control agents travel through cluster head nodes and deliver the message.

- Once the cluster head agent receives the message, it will create multiple control agents with the same message.
- Control agents travel within their clusters and ask the sensing agents to stop processing the collected data.
- Sensing agents will forward the raw data as it is collected from the environment.
- Control agents will die.

VII. SIMULATION

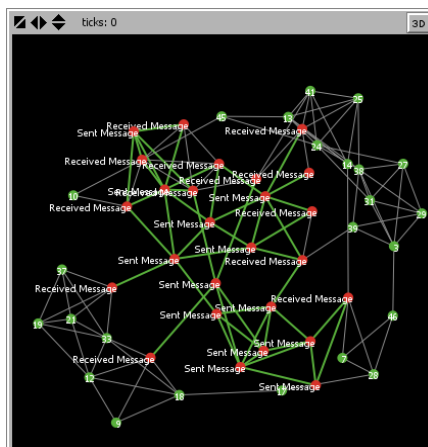
Netlog simulator is used to implement multi agents in wireless sensor network. From simulation we can conclude the benefits of using multi agents over normal client server architecture in complex systems such WSN. However client-server model consumes systems resources and communication bandwidth.

Fig. 8. Initial cluster with 50 sensor nodes



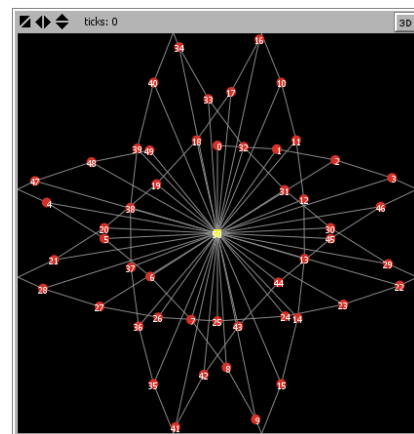
In normal client server architecture in order to send the message from the source to its destination, the source node starts flooding the message to all its neighbor. Once the message reaches the neighbors they will also resend it, this process continues until the message reaches all the nodes including the cluster head node.

Fig. 9. Flood a message in client server architecture



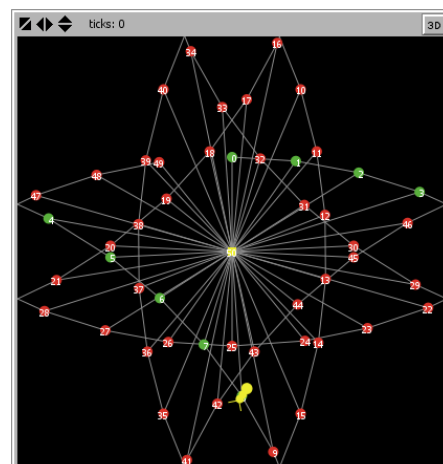
Next we show how to aggregate the data from sensor nodes using the aggregation agents with a cluster based topology, however the communication bandwidth is reserved and the data aggregate is more efficient.

Fig. 10. Initial cluster based on star topology with 50 nodes



Now to aggregate the data stored at sensor nodes and for simplicity one aggregation agent is generated and it visits the nodes one by one and collecting the data stored in their local tables. However when its payload reach the maximum size it will return to the cluster head node and provides the data then dies. A new aggregation agent is generated and it will continue with the same process until the data is collected from all sensor nodes.

Fig. 11. Aggregating data using the aggregate agent



## VIII. CONCLUSION

In this research we have proposed the multi agents paradigm for wireless sensor networks implementation, however agents are used for control, sensing and aggregating data. Complexity, scalability and adaptation of wireless sensor networks were studied in this research and a suitable solution using intelligent agents were proposed to address those aspects of such systems. In WSN where we have a large number of sensing nodes distributed over a specific area almost there are many redundant nodes which collect the same data, this type of redundancy is needed in WSN and we can consider it as a good redundancy if we have the control over the network, however we showed how the intelligent agents can do the control jobs over their clusters and put the redundant nodes in the sleep mode and wake them up when they are not still redundant. Moreover agents can interact and communicate with each other, travels through the network to achieve their tasks. We proposed four methods for power saving in this system and showed how to do that using intelligent agents. Almost agents will be the brain of the system and will do whatever needed. Comparing multi agents approach with normal models used in WSN implementation, the complexity decreases significantly because of agents specialty, scalability increases and the system is easy to adapts.

## REFERENCES

- [1] Gungor, Vehbi C., Bin Lu, and Gerhard P. Hancke. "Opportunities and challenges of wireless sensor networks in smart grid." *Industrial Electronics, IEEE Transactions on* 57, no. 10 (2010): 3557-3564.
- [2] Alam, Sahabul, and Debashis De. "Analysis of Security Threats in Wireless Sensor Network." *arXiv preprint arXiv:1406.0298* (2014).
- [3] Morris, Alexis, Paolo Giorgini, and Sameh Abdel-Naby. "Simulating bdi-based wireless sensor networks." In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pp. 78-81. IEEE Computer Society, 2009.
- [4] Fortino, Giancarlo, Stefano Galzarano, Raffaele Gravina, and Antonio Guerrieri. "Agent-based Development of Wireless Sensor Network Applications." In *WOA*, pp. 123-132. 2011.
- [5] Kuorilehto, Mauri, Marko Hännikäinen, and Timo D. Hämäläinen. "A survey of application distribution in wireless sensor networks." *EURASIP Journal on Wireless Communications and Networking* 2005, no. 5 (2005): 774-788.
- [6] Sharma, Divya, Sandeep Verma, and Kanika Sharma. "Network Topologies in Wireless Sensor Networks: A Review 1." (2013).
- [7] Krishnamachari, Bhaskar, and S. Sitharama Iyengar. "Efficient and fault-tolerant feature extraction in wireless sensor networks." In *Information Processing in Sensor Networks*, pp. 488-501. Springer Berlin Heidelberg, 2003.
- [8] Haghighi, Mo. "Cooperative Task Allocation in Utility-Based Clustered Wireless Sensor Networks." *International Journal of Information and Electronics Engineering*, Vol. 3, No. 6, November 2013
- [9] Tynan, Richard, David Marsh, Donal O'kane, and Gregory MP O'Hare. "Intelligent agents for wireless sensor networks." In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pp. 1179-1180. ACM, 2005.
- [10] Rupert, Maya, Amjad Rattrout, and Salima Hassas. "The web from a complex adaptive systems perspective." *Journal of Computer and System Sciences* 74, no. 2 (2008): 133-145.
- [11] Li, Mo, Yajun Wang, and Yu Wang. "Complexity of data collection, aggregation, and selection for wireless sensor networks." *Computers, IEEE Transactions on* 60, no. 3 (2011): 386-399.
- [12] Haque, Md Alimul, Md Faizanuddin, and N. K. Singh. "A Study of Cognitive Wireless Sensor Networks: Taxonomy of Attacks and Countermeasures." (2012).
- [13] Shin, Jaewon, AM-C. So, and Leonidas Guibas. "Supporting group communication among interacting agents in wireless sensor networks." In *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, pp. 2375-2380. IEEE, 2005.
- [14] Sangualagi, Prashant, A. V. Sutagundar, S. S. Manvi, and Vidya S. Bennur. "BDI Agents for Information Fusion in Wireless Sensor Networks." *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 1, no. 7 (2012): pp-144.
- [15] Chen, Min, Sergio Gonzalez, Yan Zhang, and Victor CM Leung. "Multi-agent itinerary planning for wireless sensor networks." In *Quality of Service in Heterogeneous Networks*, pp. 584-597. Springer Berlin Heidelberg, 2009.
- [16] Wang, Hao. "Wireless sensor networks for an extended city intelligent transportation system." *International Journal of Advancements in Computing Technology* 3, no. 5 (2011): 300-307.
- [17] Alippi, Cesare, Giuseppe Anastasi, Mario Di Francesco, and Manuel Roveri. "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors." *Instrumentation and Measurement, IEEE Transactions on* 59, no. 2 (2010): 335-344.
- [18] Momani, Mohammad, and Subhash Challa. "Survey of trust models in different network domains." *arXiv preprint arXiv:1010.0168* (2010).