International Conference on Research and Development in Science, Technology, Engineering, Management, Applied Sciences, Pharmacy, Education, Law and Humanities: The Futuristic Approach
ISSN: 2349-6002

Mobile Agents' Technology Utilization for Energy Efficiency in Wireless Sensor Network

Mohammed Abbas Qureshi[1], Dr Mohd Ateeq Ur Rahman[2]
[1]*Research Scholar, Career Point University*
[2]*Researcher Supervisor, Career Point University*

Abstract **- A wide range of high-impact applications may be supported by wireless sensor networks (WSNs), which are becoming more popular. A difficult challenge for software developers is how to deal with WSNs because of the unique properties of these networks, which combine distributed sensing, processing, and communication in a single system. Using code mobility as a distributed computing paradigm, mobile agents have exhibited excellent efficacy and efficiency in IP-based dynamic contexts. When used in conjunction with WSNs, mobile agents, in our opinion, may be more beneficial than in more traditional dispersed setups. Changing or recharging the battery might be challenging since many of these devices are located in places that are tough to reach. As the sensor data is sent to the base station and the event is sensed, the battery life of energy-constrained sensors is shortened dramatically. Routing holes and shorter network lifespans are both the result of nodes running out of energy. This research provides an energy-efficient multiple sink placement technique that maximises average network life and minimises average sensor network power consumption. For data transmission, a slew of sinks are used. The suggested approach results in a longer average network life and a lower average energy use. End-to-end latency, energy consumption, and output are all exceeded by the Protocol suggested in the simulation.**

Index Terms **– Mobile Agents technology, Energy Efficiency, Wireless Sensor Network, etc.**

I.INTRODUCTION

There has been considerable interest in wireless sensor networks as an emerging technology. a wireless sensor network (WSN) is a collection of autonomous sensors that work together to monitor ambient parameters, such as temperature or sound. Sensor networks are often self-organizing Ad-hoc systems that are made up of a large number of low-cost, autonomous sensors. An energy source, such as a battery, is frequently included in a sensor node, along with a radio transceiver or other wireless communications device. Nodes in a sensor network have radio transmission ranges that are often orders of magnitude less than the network's overall geographic span.

Data collection in Wireless Sensor Networks has recently been suggested to use a mobile agent (MA), a component of the mobile computing paradigm (WSNs). Two methods are used in the MA-based technique to acquire data more efficiently: Single-agent itinerary planning (SIP) and multi-mobile agent itinerary planning (MIP). The use of distributed multi-MAs to carry out the data collection work was recommended in the MIP to overcome the limitations of SIP. There is still a lot of debate on how many MAs and their routes to use, notwithstanding the benefits of MIP, Interest in mobile agent systems for high-level inference and monitoring in a wireless sensor network has grown recently (WSN). Flexible re-tasking of applications, local processing, and collaborative signal and information processing are all made possible by mobile agent systems that use migrating codes. In contrast to traditional WSN operations based on the client-server computing architecture, this offers more flexibility and additional capabilities to WSNs.

II. MOBILE AGENT TECHNOLOGY

A mobile agent is a program functions on behalf of a user in a distributed environment and is able of migrate independently from one node to other to accomplish the assigned task of user.

Mobile agent = agent + mobility

Software agents and distributed computing are used to create a mobile agent. By reducing network traffic, mobile agents may also help to alleviate network latency.

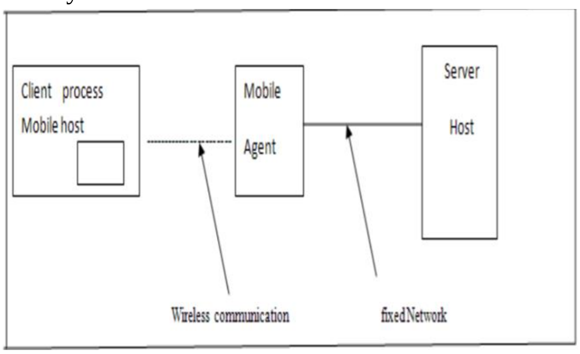

Figure 1: Mobile Agent Technology

Figure 1 depicts the use of mobile agents to facilitate communication between a server and a client host. When we talk about 'mobile agents,' we're referring to software entities that are self-aware enough to be able to move between different nodes in a network. You may move a mobile agent around to locate what you need. Wireless technology is being used to keep them in touch.

WSNs benefit from the MA-based computing paradigm because of the following reasons [11,12]:

- Extensibility and task adaptability: Within the same network, many MAs might be allocated to distinct duties. It is possible to employ each MA to do a certain activity, which therefore allows for a variety of diverse applications. MA's task flexibility and extensibility make WSNs more useful.
- Local data processing: In order to complete an assigned job on behalf of the MA's dispatcher (sink), 'the MA may migrate from node to node and perform local processing at the node before returning to the sink.' As a result, the network's bandwidth would be reduced since the nodes would no longer have to send their data to the sink as often for processing.
- Reliability: As long as the network connection is active, the MA may be sent and will return its findings after the network connection has been reestablished. MA's performance is unaffected by network dependability, as a result
- Progressive accuracy: Nodes that have previously been visited by the MA are taken into consideration during migration. When MA moves from node to node, accuracy of the integrated output improves with time since it is dependent on information gained. Because of this, the MA may stop its migration and provide the findings when the integrated result meets a certain accuracy criterion. Using MA, the network bandwidth and calculation time are both reduced as a result of not contacting any superfluous nodes.
- Fault-tolerance: MA's route may be changed dynamically based on the information gained and energy restrictions [6]. Before deciding to migrate to the next hop, the MA might review the information of the next hop nodes. A dead node or a broken connection may be prevented here, protecting the MA from being lost because of fault-tolerance.

2.1 Work processing of Mobile Agents

First time a node is visited the mobile agent saves its code there and sends the result back to the sink node, without carrying its code. As part of the first round, the mobile agent replicates its processing code to the memory of each node. All of these nodes delete the processing code after the job is complete. When the agent gets data, it performs the filtering function. For example, if node 1, node 2, and node 3 are all within a few feet of each other, they must all have the same temperature. According to our assumptions, node1 reports a temperature of 30 degrees, but node3 reports a temperature of 30 degrees and 2 degrees. between node1 and node3, node2 is situated. If node2 is to function properly, it must have a temperature between 30 and 32 degrees. The only exception is if node2 reports a temperature of more than 40 degrees or less than 10 degrees. Fault data may be gathered through node sensors. In this approach, the data may be evaluated by agents. Mobile agents may save their current state in secondary memory when no network connection is available and then migrate to the home network when a connection is available. There is no need for software agents to travel across a network in order to connect with information sources or other agents. They're working on systems that allow agents to exchange messages.

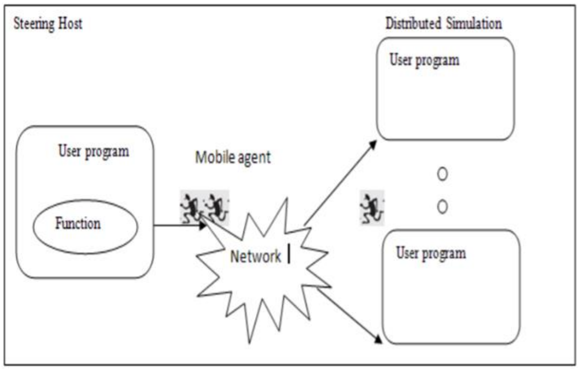

Figure 2: Clustering of Mobile Agents

User programs connect with one other through mobile agents in the network shown in fig 2. Clusters of mobile agents form here. In this system, all user programs are wirelessly connected to a specific network.

III. PROPOSED SYSTEM

3.1 Controller

As with data networks, control networks are often tiny. Packet latency and delivery must be monitored in order for the network to function. Late and lost packets have a negative impact on network performance. 'As a result, the routing information may be used to regulate and configure the right route controller support.' Trace files may be generated, packets can be sent, and data can be protected. For this authorization, the system and protocols behave and communicate in a similar way. The network serves as the hub for all operations. The networks have configured the settings of each node and are now sending packets through the system.

3.2 Network Model

Wireless sensor networks with sensor nodes V, sink locations Sp, and E connections (E stands for communication between sensors) are considered in this paper. This square area has a random distribution of sensor nodes. After deployment, we assume that all sensors and sinks will remain unchanged. It's clear that each sensor and sink has a distinct location, which is recognised. Ids range from 1 to |V| for each node and include a transmission set for that node (R). We take it for granted that all nodes have a same range of transmission. Only when the two sensor nodes u and v are within each other's range are they joined together via a connection. There will be a connection between the sink and the node if the distance between them is reduced. The power of the sinks is unrestricted. In contrast, each sensor u node is simply limited by the power RE (u). In this example, we'll assume that fu is the rate at which sensor u sends data to the sink. Because only data is sent to and received from the sink, each node is responsible for sending and receiving data.

3.3 System Analysis

As a way to save energy, we recommend increasing the average network length and reducing average power usage in a network system. There were four steps to the proposed algorithms: deployment, implementation, prospective LS sink locations, and final sink locations. Nodes and sinks will be dispersed at random across the deployment region. 'The first sink sites and nodes are aggregated during the setup phase.' During the third phase, local sinks are examined in an effort to maximise or decrease the objective function, respectively. The next step is to pinpoint the exact placements of all of the sinks. Second and third stages are carried out in each iteration to find the optimum answer.

Setup Phase: This process comprises four sub-phases, including (i) random choice of sink locations, (ii) node clustering, (iii) energy-aware generation of communication tree and (iv) average network lifetime and (v) average network power consumption estimate.

Node Deployment: Another element that affects the routing protocol's effectiveness is the topological use of nodes. Deployment might be predetermined or completely random. Automated sensors are positioned manually and their data is relayed via specified paths. In the self-organizing system, the sensor nodes form an ad hoc network despite their random placement.

Grouping of Nodes: In order to narrow down the list of potential sites, the next step is to group the sensors together. In WSNs, grouping is critical for the administration of local nodes and the balancing of nodes in terms of various quality of service (QoS) characteristics. V nodes are partitioned into groups of K=|Sp| nodes using a distance-based technique. To build a group, a node must be aware of Sp's position. Each node u is connected with a sink based on its distance from each sink siSp. In our work, we use the time of arrival (ToA) approach to calculate distance. Asi is the set of nodes closest to the sink, and a connection (u,si) is created for each uAsi pair.

Random Selection of Sink Locations: We choose K locations from set of sink locations SG on a random basis at each iteration Sp, where Sp ≤ SG, is the candidate sink location.

IV. RESULT AND DISCUSSION

We use the NS-2.35 simulator for our investigations. The trials are divided into two sections. The first step is to make sure our strategy is feasible and to look at the delays, energy usage, and performance in more depth. There are now 43 mobile nodes in the network, and communication may commence between the source and destination. A single node's position may be used to estimate the distance between two points. Number of data flows per server-to-device interaction estimated. Each node's transmission rate will be determined by the amount of time it takes to run the simulation. As part of our job, we are able to retain resources, postpone communications, and determine the best route for each node to take. In a radio broadcast of 250 metres, we employ CBR network traffic at 1000 bytes/0.1 milliseconds. Application traffic is what we utilise. Sending data is done in the form of 1000-byte packets, which have a transmission rate of 2Mbps and a maximum speed of 25 metres per second. The simulation duration is 10s, the network area is 1000x1000, and AODV is used to route it. Among the many routing methods are RPA, DMADA, and EEMSP, to name just a few.

Table.1. Simulation Table

Parameter	Value
Transmission rate	1000 packets/0.1ms
Application Traffic	CBR
Packet size	1000 bytes
Radio range	250m
Maximum speed	25m/s
Channel data rate	2Mbps
Number of nodes	43
Simulation time	10s
Routing methods	RPA, DMADA, EEMSP
Routing protocol	AODV
Area	1000×1000

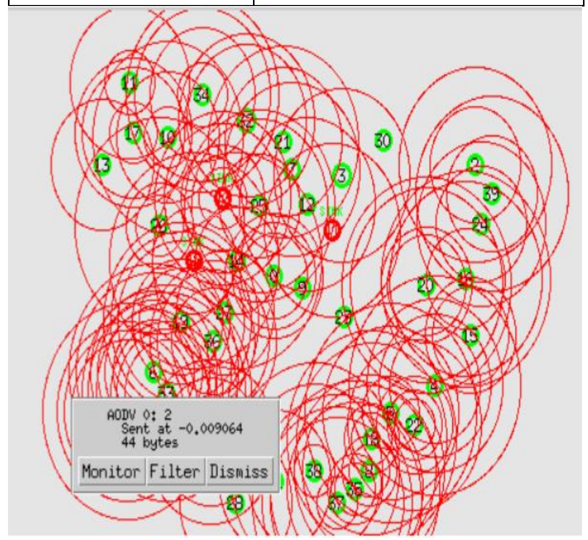

Figure 3: Network Deployment with Broadcasting

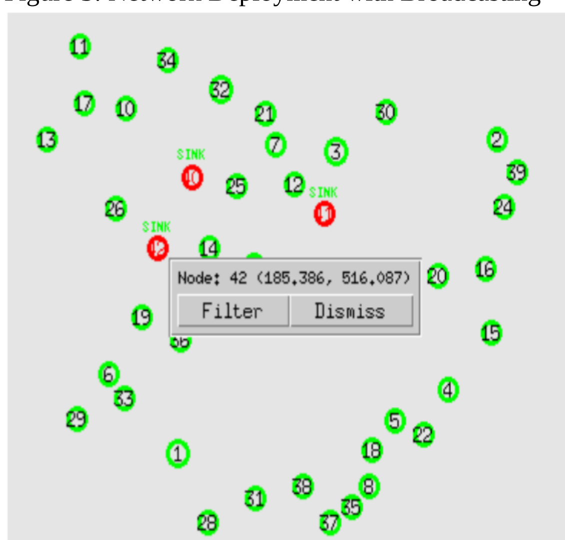

Figure 4: Hello Packets Exchange in network

Figure 3 depicts the network's node deployment. Node topology values and the name window's characteristics are shown in the display. The network's broadcast for communication may be seen in this screenshot. Participation by all nodes is required for this procedure to work properly. All nodes exchange packets with the aid of the routing protocols. Figure 4 shows the value of each topological node. Each node in the network is placed at a random location via the random waypoint model.

```
link.tcl

set xx  [expr rand() * 300]
set yy  [expr rand() * 900]
$ns_ at 2.0 "$node_(40) setdest $xx $yy 500"
puts  "################################"
puts  "The SINK1 positions are : $xx $yy"
puts  "################################"

set xx1  [expr rand() * 600]
set yy1  [expr rand() * 900]
$ns_ at 2.0 "$node_(41) setdest $xx1 $yy1 500"
puts  "################################"
puts  "The SINK2 positions are : $xx1 $yy1"
puts  "################################"

set xx2  [expr rand() * 900]
set yy2  [expr rand() * 900]
$ns_ at 2.0 "$node_(42) setdest $xx2 $yy2 500"
puts  "################################"
puts  "The SINK3 positions are : $xx2 $yy2"
puts  "################################"
```

Figure 5: Sink positions displayed

The sink locations in the network are shown and illustrated in Fig.5 above. The location of sinks is determined by the destination's random number and address. Any additional data to be sent is determined by the position of the sink node.

```
File  Edit  View  Search  Terminal  Help
Distance from node(31) --to--node(20)------------>302.78397243128506
Distance from node(31) --to--node(21)------------>766.81394123084726
Distance from node(31) --to--node(22)------------>751.7707231745386
Distance from node(31) --to--node(23)------------>406.94494004793455
Distance from node(31) --to--node(24)------------>347.64408839929257
Distance from node(31) --to--node(25)------------>606.94955983545867
Distance from node(31) --to--node(26)------------>454.60397852500608
Distance from node(31) --to--node(27)------------>551.78582176772136
Distance from node(31) --to--node(28)------------>366.9573167853402
Distance from node(31) --to--node(29)------------>638.77457077071017
Distance from node(31) --to--node(30)------------>106.46510200112874
Distance from node(31) --to--node(31)------------>0.0
Distance from node(31) --to--node(32)------------>450.47019685505177
Distance from node(31) --to--node(33)------------>486.4428462110842
Distance from node(31) --to--node(34)------------>165.25146434111608
Distance from node(31) --to--node(35)------------>227.0618413545177
Distance from node(31) --to--node(36)------------>880.61109518290868
Distance from node(31) --to--node(37)------------>659.42907930631327
Distance from node(31) --to--node(38)------------>609.48309659299798
Distance from node(31) --to--node(39)------------>689.89259942040212
Distance from node(31) --to--node(40)------------>548.64737053929787
Distance from node(31) --to--node(41)------------>541.71919450439225
Distance from node(31) --to--node(42)------------>639.10008521383884
Distance from node(32) --to--node(0)------------>790.70441932945403
Distance from node(32) --to--node(1)------------>395.34611923059856
Distance from node(32) --to--node(2)------------>170.55795664030958
Distance from node(32) --to--node(3)------------>267.18549712447196
Distance from node(32) --to--node(4)------------>127.32559432036132
Distance from node(32) --to--node(5)------------>183.88257661240576
Distance from node(32) --to--node(6)------------>367.79293408990469
Distance from node(32) --to--node(7)------------>751.67978712593754
```

Figure 6: Distance between users

In Fig.6, the distance between each node and the next is displayed. The Euclidian formula is used to calculate the distance between users. Figure 7 depicts the delay as well as the simulation time as a function of time. 'There are numerous SINKs performed, which lessen the delay in time by minimising the contact time between nodes compared to older techniques like RPA, DMADA and EEMSDA.' It's possible to see the simulation time as it relates to throughput in the graph shown in Figure 7 (above). An increase in throughput is seen as compared to earlier approaches like RPA, DMADA, and EEMSDA.

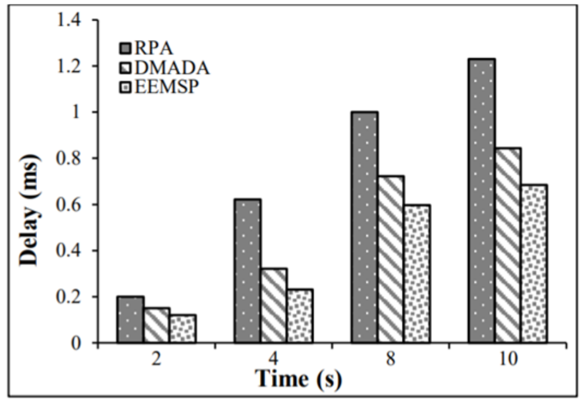

Figure 7: Routing Delay

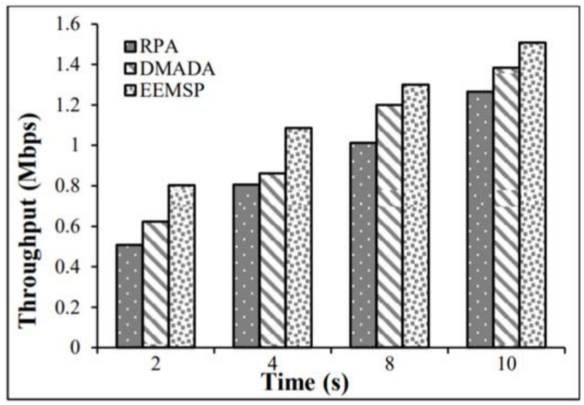

Figure 8: Network Performance

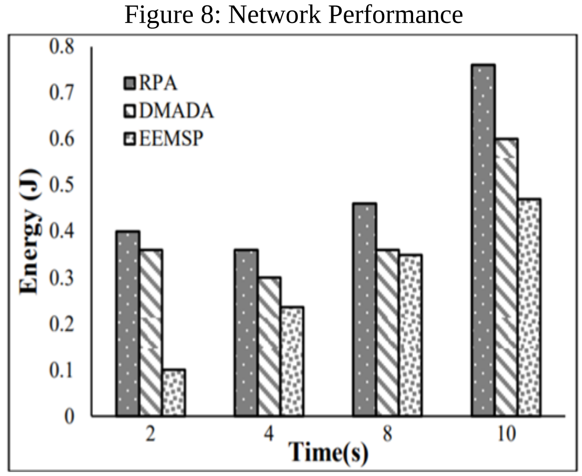

Figure 9: Energy Consumption

The Fig.9 above shows the energy consumption and time vs. energy simulation, as shown in the figure. In comparison to prior approaches like RPA, DMADA, and EEMSDA, the Multiple SINK system decreases energy values.

V. CONCLUSION

Because sensor nodes have limited resources, ensuring that communication is done in an efficient manner is a top priority. Sensors with limited power consumption rapidly consume their batteries when they transmit sensor data to the base station while detecting an event. For the most part, the suggested method focuses on transmitting sensed data via static sensor nodes and multi-hop routing. Because it is inefficient and prone to error, multi-hop communication in WSNs is an outdated technology that should be phased out. There are a number of ways to locate WSN energy efficiency sinks in this study. Our ultimate objective is to reduce overall network energy consumption while simultaneously increasing overall network energy consumption. In order to improve network energy efficiency, we present a new method that executes a restricted number of repetitions. Data aggregation from Dynamic Mobil Agents and the random placement method is used to replicate algorithmic proposals. Experiments have shown that the algorithms described outperform those already in use. We utilise the NS-2 simulation program to model the network process.

REFERENCE

[1] A. H. Sodhro, L. Chen, A. Sekhari, Y. Ouzrout and W. Wu, ‘Energy efficiency comparison between data rate control and transmission power control algorithms for wireless body sensor networks,’ International Journal of Distributed Sensor Networks, vol. 14, pp. 1550147717750030, 2018.

[2] A.R. Silva, A. Romao, D. Deugo, M. Mira da Silva, ‘Towards a Reference Model for Surveying Mobile Agent Systems,’ Autonomous Agent and Multi-Agent Systems, 4(3), pp.187–231, 2001.

[3] C-L. Fok, G-C. Roman, C. Lu, ‘Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications,’ Proc. of the 24th Int’l Conf. on Distributed Computing Systems (ICDCS'05), Columbus (OH), Jun 6-10, pp. 653-662, 2005.

[4] D.B. Lange and M. Oshima, ‘Seven Good Reasons for Mobile Agents’, Communications of the ACM, 42(3), 1999.

[5] G. Fortino, A. Garro, S. Mascillaro, W. Russo, ‘Specifying WSN Applications through Agents Based on Events and States,’ Proc. of the IARIA/IEEE Int’l Conf. SensorComm’07, Valencia, Spain, Oct. 14-19, 2007.

[6] G. Mehmood, M. Z. Khan, H. U. Rahman and S. Abbas, ‘An efficient and secure session key establishment scheme for health-care applications in wireless body area networks,’ Journal of Engineering and Applied Sciences, vol. 37, no. 1, pp. 9–18, 2018.

[7] L. Szumel, J. LeBrun, and J.D. Owens. ‘Towards a Mobile Agent Framework for Sensor Networks.’ Proc of 2nd IEEE Works. on Embedded Networked Sensors (EmNetS-TT), Sydney, Australia. May 30-31, 2005.

[8] M. Nabi, M. Geilen and T. Basten, ‘MoBAN: A configurable mobility model for wireless body area networks,’ in Proc. of the 4th Int. ICST Conf. on Simulation Tools and Techniques, Barcelona, Spain, pp. 168–177, 2011.

[9] O. Albahri, A. Albahri, K. Mohammed, A. Zaidan, B. Zaidan et al., ‘Systematic review of realtime remote health monitoring system in triage and priority-based sensor technology: Taxonomy, open challenges, motivation and recommendations,’ Journal of Medical Systems, vol. 42, no. 5, pp. 1–27, 2018.

[10] S. González-Valenzuela, M. Chen and V. C. Leung, ‘Applications of mobile agents in wireless networks and mobile computing,’ Advances in Computers, vol. 82, pp. 63–113, 2011.