Hybrid Ant Colony Optimization for Sensor-Cloud Network

Yogita S. Baviskar[1], Dr. Shailaja C Patil[2]
[1]M.E, ENTC Department, JSPM's RSCOE, Pune, India
[2] Prof. ENTC Department, JSPM's RSCOE, Pune, India

Abstract* - Wireless sensor network (WSN) consists of a large number of sensor nodes that are interconnected to form a wide communication network. Usually, it can achieve small size, low cost, low power consumption, fewer network components and other features easily. With the development of hardware limitations, better performance and enhancing greater computing capability, people turn to find other techniques to achieve these goals. Therefore, the concept of "Cloud" was born. In fact, as early as the Internet appeared, the "Cloud" has already existed silently providing for us some services. A wide use, the fast increasing data emanating from WSN is not fully utilized due to the limitation for structure of WSN. In recent years, the "Cloud" concept has become more and more popular, especially on the business sector. There are also many types of cloud computing platforms such as Google, Amazon, IBM, and Microsoft etc. In this project, provides load balancing in cloud based wireless sensor network which will be used for dividing the processing work and delivering faster services with higher efficiency. It will help to improve the performance and retain the system stability.

***Keywords:* Wireless Sensor Network, Hybrid Ant Colony Optimization, Virtual Machine.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are used in many application areas including health, agriculture and gaming. Wireless Sensor Networks (WSNs) are combinations of micro-electro-mechanical systems, wireless communication systems and digital electronics nodes that sense, compute and communicate. Up till now the real world deployments of WSNs have been tailor-made solutions where applications are bundled with a WSN at the time of deployment with no possibility for other applications to re-use the deployed WSN. In the past, applications of sensor networks were thought to be very specific. The communication protocols of sensor networks were also very simple and straightforward. Some researchers were even against the use of the compatible in networking protocol architecture in WSNs [1]. There were different reasons behind that such as the resource constraints for layered architecture, the problems of configuring large number of devices, and the essence of sensor nodes' distinct identity. But with the advent of the Internet of Things [2] and federated IP-WSNs, this demand is going to be blurred.

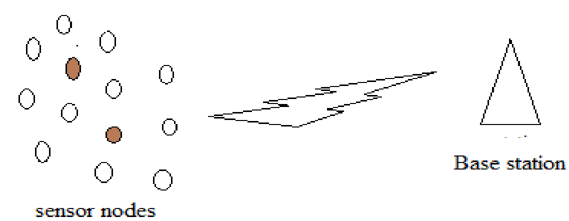

Fig. 1. Wireless sensor network

The huge numbers of IPv6 addresses, the necessity for end-to end communication and advances in microelectronics have changed the concepts of the research community. Now, a tiny sensor node can hold a compatible TCP/IP protocol stack, so we can now think of using the concept of networking protocols in IP-WSNs. services to user. The fundamental of cloud computing consist of distributed and grid computing. Cloud computing consist of three services which are as follows: Infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) [6]. In an IaaS framework, the services that are provided by cloud computing are similar to providing hardware resources to users, such as in Amazon S3 [4] and Amazon EC2 [3]. Virtualization is one of the most important concepts in this type of cloud computing.

II. RELATED STUDIES

The main difference between scheduling in a cloud environment and traditional single processor scheduling is that there are several processors in a cloud system. Additionally, different processor units can contain different resources in a cloud, and thus, the cost and execution time to serve the same task in different processor units could also be different. When scheduling in a cloud system, we must consider the cost and performance of different processor units. This concern is analogous to multi-core scheduling [12] and scheduling in grid computing [21]. Typically, scheduling in a multi-core processor or a cloud computing system can be divided into two parts: the first part is to determine the execution order of the tasks, while the second part is to determine which task should be dispatched to which processor unit. Task scheduling and mapping with different limitations is an important issue in multi-processor systems. This type of optimization problem is an NP-complete problem [12]. In [21], there are three task queues in a system, which represent high priority tasks, medium priority tasks and low priority tasks. When requests arrive, the algorithm calculates their priorities according to the costs of those requests on each processor and then dispatches them to the corresponding task queue. Finally, the system executes the tasks according to their priorities. Lin and Lu [10] proposed the SHEFT method. In this method, after all of the tasks are prioritized, SHEFT chooses the task that has the highest priority and calculates its finish time in each of the processor units, and then, it dispatches the task to the processor that has the earliest finish time. In addition to the methods mentioned above, some studies have recently used heuristic algorithms for scheduling in cloud computing systems. The purpose of meta heuristic [5] is to find a feasible solution in a limited amount of time. This type of method uses strategic guesses to find plausible directions or combinations from an enormous solution space and, through repeated calculations, locates a better feasible solution. The meta-heuristics are composed of three main procedures, namely permutation, evaluation and determination. Most studies that have used meta-heuristics in scheduling were applied to large distributed systems, grid computing [17] and job shop scheduling problems [14], including genetic algorithms [23] ant colony optimization (ACO) and particle swarm optimization (PSO) [16]. In these studies, ACO could be the most suitable method for solving job shop scheduling problems. The most important factor is that ACO is not very sensitive to the initial

III. SYSTEM MODEL

Network Model: In this research, a sensor network consisting of N sensor nodes which are randomly deployed over an environment. Sensor nodes collect the sensing data from the surrounding environment and send these collected data to BS.

Sensor Nodes and Energy Consumption Model: Sensors are usually wireless electronic elements and functions for sensing interactions. Sensors are located in the environment for data-gathering purposes (such as temperature, sound, vibration, pressure, motion, or pollutants) from the monitored area [17]. In recent years, with progress in technology, the wireless devices have been smaller, cheaper, and more powerful. Wireless sensor networks (WSNs) consist of sensor nodes that are designed with special purposes and applications (scientific, monitoring, or military purposes). In WSN, nodes are usually homogenous and consist of some units such as battery, sensors, transceiver, processor, and memory [18].

The nodes send the collected data to the base station (BS). The BS is responsible for collecting data sent by the other nodes usually located in the center of the environment and sometimes has different units in comparison with the other nodes. Due to its interaction with other nodes and in some cases local data processing, the BS has power source with greater energy, larger memory, or maybe stronger processor. In the real system, all units of each node cooperate in doing delegated tasks that has effect on node energy consumption and in general on the network energy [26]. It should be noted that each node has a limited and irreparable energy source and if a node is turned off due to the completion of power source, it will reduce the connectivity degree and in some cases fragmentation in the network.

Sensor Nodes Behaviour: In general, each node is placed in one of the two mechanisms based on the current state: active or sleep. In the active mechanism, a node uses efficient protocol on the network energy instead of turning off the transceiver for saving energy, while a node in the sleep mechanism has no interaction with other network nodes due to the fact that its transceiver is turned off and that the node energy consumption is lower [15].

In the active mechanism, each node will be in one of the three operational modes: transmit, receive, or idle. In the first mode (transmit), more node energies are consumed for turning on the transceiver and packet’s transmit. In the second mode (receive), the node with its transceiver turned on receives a packet, demodulation, and decoding [19] where these operations (packet processing and turned on transceiver) cause energy consumption in the node. After the packet’s receiving or sending operations, a node is placed in the idle mode. In the idle mode, each node listens to the communication channel without any sending or receiving in an active manner. In this mode, some functions in the hardware can be switched off, but all circuits are maintained to be ready to operate.

AODV Routing Overview: Ad hoc on-demand distance vector (AODV) is one of the most used reactive routing protocols for WSN. In the reactive routing, paths are determined only when needed [13]. The primary objectives of this protocol are (a) to broadcast discovery packets when needed, (b) to distinguish between local connectivity management (neighbourhood detection) and general topology maintenance [21], and (c) to propagate information on node connectivity degree for its neighbours or other interest nodes. For example a node which is aware of its surrounding environment (e.g., neighbour nodes) locally broadcasts a HELLO message; also the route request (RREQ) packets are sent if a sender is finding a route to BS. In this case, the path is made by route reply (RREP) packet uni casting to sender.

The AODV uses the route discovery mechanism with broadcasting instead of the source routing. Each node has a local routing table (RT) for quick response time to requests and establishment. Each row of RT shows the next hop from this node to the destination. The route discovery process is implemented when a node needs to communicate with other nodes and route information does not exist in its RT. The protocol uses the sequence number for more maintenance of the routing information among nodes. This sequence number will cause the efficient use of network bandwidth by minimizing the network load for the control and data traffic. When a node wishes to send data to the BS, the source node creates a RREQ packet.

This packet contains the source node's IP address, source node's current sequence number, the destination IP address, and destination sequence number that are broadcast in the source transmission range. Broadcasting is done via flooding. Finally, this packet will receive a node that possesses a current route to the destination. Firstly, the receiver checks the RREQ packet. If the intermediate node has an entry in RT for the desired destination, the intermediate node will make a decision by comparing its entry sequence number with RREQ packet sequence number.

If the intermediate node has not received this RREQ before, meaning that it is not the destination and does not have a current route to the destination and RREQ sequence number is bigger than saved sequence number, it rebroadcasts the RREQ [11]. When the intermediate node is capable of replying, it means that it has a sequence number greater than or equal to that contained in the RREQ. If a node receives a packet with the same broadcast ID and source address, it drops this RREQ packet (sometimes maybe a node receives multiple copies of the same RREQ packet) [14]. Once an intermediate node receives a RREQ and its destination, the node sets up a reverse route entry for the source node in its RT. Reverse route entry consists of source IP address, source sequence number, number of hops to source node, and IP address of node from which RREQ is received. Using the reverse route, a node can send a RREP packet to the source. The RREP is uni-cast in a hop-by hop model to the source [17]. While the PREP backward to the source, each intermediate node creates a route to the destination and sets up a forward pointer to the node from which the RREP comes.

MAC Layer: To avoid collision, the medium access control (MAC) protocols have developed and control how each node could access the channel. IEEE 802.11 is a distributed MAC scheme that works based on the carrier sense multiple accesses with collision avoidance (CSMA/CA). In this scheme, each node accesses the medium on a contention basis. Before a data transmission begins, the sender and receiver must have a RTS-CTS signalling handshake to reserve the channel [12]. A sender for sending a packet, firstly, with its transceiver senses the channel and looks up its network allocation vector (NAV). Therefore if the channel is free, it sends a RTS (request-to-send) to the destination node, and the destination with CTS (clear-to-send) approves RTS receiving; otherwise it has to wait until the channel is busy.

Traffic Protocol: The constant bit rate (CBR) service category is used to transport traffic connections at a constant bit rate, where there is an inherent reliance on the time synchronization between the traffic source and destination. The consistent availability of a fixed quantity of bandwidth is considered appropriate for the CBR service. Cells which are delayed beyond the value specified by the cell transfer delay (CTD) are assumed to be significantly of less value to the application. In the CBR, bandwidth guarantees the peak cell rate of the application.

IV. PROPOSED METHODOLOGY

Hybrid Ant Colony Optimization Model

When a node wishes to send collected data to BS it makes a RREQ packet and broadcasts it to all neighbours. We used two kinds of ants: forward and backward ants that have been presented by FAnt and BAnt. RREQ packet is used for modelling the FAnt and BAnt. In this research, we have several FAnts (depending on the source neighbour's number) and a BAnt that will be discussed in the next sections. FAnt tries to find a route from the sender to the BS (a FAnt is launched from the source and intermediate nodes until the BS).While FAnt moves forward, it saves the list of nodes that has been visited in its memory and tries to avoid travelling the same node and BAnt fixes this route. In our algorithm, we restricted and targeted the number of RREQ packets (FAnts) sent by each node in compared with the original AODV. The winner FAnt makes a BAnt with the same parameters and BAnt returns to the source and updates RT and pheromone concentration amount for each node in its path. We used the pheromone rules the same way as in [16].

BAnt when traversing from a node will increase the pheromone, where α is the variable parameter, hm represents the maximum hops between sources to destination, hc is the remaining hops to destination, and E_n represents node remaining energy. We added the node remaining energy to this equation based on the reason that the ants select the node with greater power.

$$\Delta\tau = \alpha(hm - hc)E_n$$

When the node with identifier $\Delta\tau$ transmits the packet, it updates the pheromone concentration amount with the following equation:

$$\tau_n = (1 - \rho)(\tau_n)(\Delta\tau)$$

Where ρ is the pheromone evaporation (pheromone amounts evaporate very soon when no ants traverse from this path) coefficient, the range of ρ is [0, 1], and 1- ρ is the pheromone residue factor. We added a structure to FAnt and BAnt to save the travelling path in addition to the original fields which used AODV for RREQ and RREP packets.

Energy Optimized Parameters

If FAnts act intelligently in choosing the next node, we will be nearer to our aim. We aim to assist FAnts in choosing the next node. By this method, the FAnts, in addition to the pheromone concentration amount for choosing next node, will use the extra parameters, where these parameters contribute to the more intelligent choices. In this proposal, we focus our attention on factors such as remaining energy of each node as well as the connectivity degree of the same node, the distance of node to its neighbours, and then its pheromone concentration amount. These factors are briefly discussed as follows.
(i) Remaining Energy. We know that each node has a battery with limited power, and energy saving for batteries is essential for node survival and network maintenance. On the other hand, as we will select the node with the highest remaining energy, the network lifetime will increase. We assume impact factor 1.5 for the remaining energy.
(ii) Connectivity Degree. The numbers of node neighbours which are in the transmission range of the node are defined as the connectivity degree of the node. This is important because if a node with a higher degree is elected, firstly, with higher probability we can find the destination with this node and, secondly, perhaps the destination is one of its neighbors that can decrease the length of the path as well as reducing the energy consumption of the whole network. Also, we use impact factor 1.5 for connectivity degree.
(iii) Distance of Node. As the distance between two nodes within the transmission range grows, they need more energy to send/receive data, but with high probability the destination is nearest to this node. The distance between the current node to this neighbour calculates with (8) that in this formula x, y shows the location of current node and x_n, y_n is the location of the destination (here one of current node neighbour).This list has a row for each current node neighbour. We use the impact factor 1 for it.

Linguistic Values and Membership Functions
It shows, each factor is being presented by three linguistic values. Linguistic values, namely, low, medium, and high, are used for input parameters of the remaining energy, connectivity degree, and pheromone concentration amount. The use of near, moderate, and far for distance; also the values of very bad, bad, good, and very good are used for output parameters. Hybrid Ant colony optimization is used to reduce a large computation time which is used in scheduling.
Request rejecter operator : Here, on each server how much memory is remaining is check by the algorithm, and calculate maximum amount of remaining memory. When remaining memory is less than incoming requests memory, then that request will be rejected. For example, considering two servers are in the system, X1 and X2. The remaining memory in X1 is 2 GB and in X2 is 5 GB. Now, two requests y1 and y2 are entering in the system, and the demand of their memory is 1 and 7 GB. When new request arrives, reject operator will checks the maximum remaining memory in one server, which is 5 GB, and then check if each new coming request is possible being accepted. Since the memory demand of y1 is 1 GB, smaller than the maximum remaining memory in one sever, representing that y1 is possible being serviced by any server. The memory demand of y2 is 6 GB, bigger than the maximum remaining memory in one sever, representing that there is no sever containing enough memory to serve y2. As consequence, y2 will be rejected by reject operator, and only y1 needs to be scheduled this time.

Search operator: Search is used to construct the solutions for all of the ants. Each ant will select the next path according to the probability Matrix.

V. SIMULATION AND RESULTS

The performance of algorithm is evaluated by network simulator 2 (NS2) and is implemented in two scenarios. NS2 is one of the most famous and most widely used network simulators. In this simulation, we consider a network with 10, 20, 30, 40, 50, 60, 70, and 80 nodes that are randomly placed in a dimension of 1500M * 1500M area. Each simulation is running for 180 seconds of the simulation time. Here implemented two scenarios based on the node transmission range (100M transmission range for scenario 1 and 300M transmission range for scenario 2) and repeated the simulation for different number of nodes in each scenario. Table 1 lists the simulation parameters. All nodes were taken randomly in the environment and controlled the hop count in the simulation time. Different parameters for simulation are shown by Table 1.

Table 1. Simulation Parameters

Parameters	Values
Channel type	Wireless
propagation model	TwoRayGround
network interface type	WirelessPhy
interface queue type	Drop Tail
antenna model	Omni Antenna
max packet in queue	50
number of mobile nodes	22
routing protocol	AODV
X dimension of topography	1800
Y dimension of topography	840

The amount of energy consumption in the network depends on the amount of energy required to transmit a message from a sender to a receiver is shown in fig 1, if a proposed algorithm is able to reduce energy consumption in the network, the network lifetime will increase.The numerical values that used for both scenarios are illustrated in Table 2. There are many effective parameters on energy consumption in the network such as network setup time, routing setup time, CPU processing, transmitting packets, and receiving packets, and here measure the final energy consumption for the network at the end of simulation time.

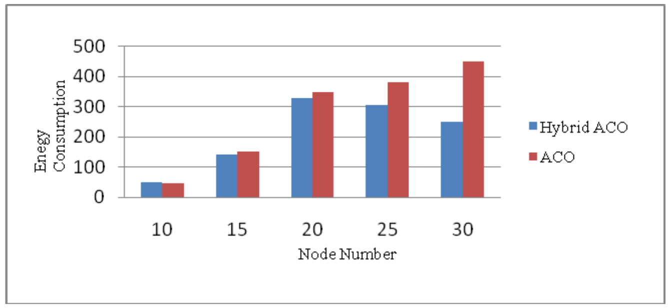

Fig.1. Energy Consumption

5.1 Average End-to-End Delay

The average end to end delay is shown in fig 2. The execution time is compared in table 2. This is shows that hybrid ACO algorithm, although it has a bigger routing setup time for finding optimal path, and when the path was determined, the optimal path of the end-to-end delay or the packet delivery time was better than the original ACO. Paying heed to this point, this is solving the increased routing setup time issue, because the energy will consume more when sending and receiving packets.

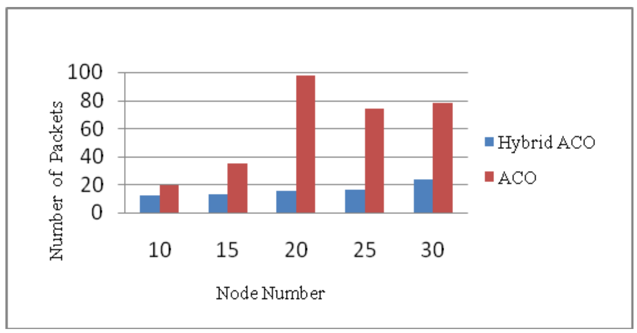

Fig. 2. End To End Delay

Table 2. Execution TIME

Test Data	Execution time (s)	
	ACO	Hybrid ACO
1	3.5129	1.2511
2	4.3423	2.4288
3	8.6729	3.4556

In summary, all the above results indicate that our proposed method performs better than the traditional method. Therefore, this method is efficient to be used in WSNs because it balances the load between gateways and the remained energy among sensors. In addition, this method reduces not only the system energy cost but also the packet loss ratio and the average packet delay.

VI. CONCLUSION

This paper proposed a hybrid ACO approach for VM scheduling with load balancing. Hybrid ACO can serve customized VM requests for which resources are considered for load balancing. Furthermore, the server status information, which is the workload of historical requests, is used to predict the workload of the new input requests. Load balancing help the network for utilizing the available resources optimally, minimizing the resource consumption to improve the overall performance of the system. Load balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio, making sure that no single node is overloaded. Simulation results show the correct operation of the protocol and its suitability to be used in a wide range of applications and scenarios. The performance of this proposal has been compared with the ACO. This algorithm increases the network lifetime by reducing the nodes energy consumption and the number of packets.

REFERENCES

[1] Marino, Claudio S., and Paul M. Chau., "Constrained Sensor System Resolves Closely Spaced Obscured Low-SNR Sources Using Virtual Beam forming." IEEE Transactions on Instrumentation and Measurement, Vol. 63, No. 1,pp. 231-242, January 2014.

[2] Dhanapala, Dulanjalie C., and Anura P. Jayasumana. "Topology Preserving Maps Extracting Layout Maps of Wireless Sensor Networks From Virtual Coordinates." IEEE/ACM Transactions on Networking, Vol. 22, No. 3, pp. 784-797, June 2014.

[3] Dong, Jing, et al. "Secure and robust virtual coordinate system in wireless sensor networks," ACM Transactions on Sensor Networks, Vol. 6, No. 4, Article 29, pp. 1-34, July 2010.

[4] Winkler, Thomas, and Bernhard Rinner. "Security and Privacy Protection in Visual Sensor Networks: A Survey." ACM Computing Surveys, Vol. 47, No. 1, Article 2, pp. 1-42, April 2014.

[5] Mahmood, A., et al. "Home Appliances Coordination Scheme for Energy Management Using Wireless Sensor Networks in Smart Grids." Procedia Computer Science, Vol. 32, No. 3, pp. 469-476, 2014.

[6] Dasgupta, Kousik, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mandal, and Santanu Dam. "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing," Procedia Technology, vol. 10, pp. 340-347, 2013.

[7] Zhiqiang Wei; Jing Li; Yongquan Yang; Dongning Jia, "A residential gateway architecture based on Cloud computing," IEEE International Conference on Software Engineering and Service Sciences (ICSESS), vol., no.6, pp.245,248, 16-18 July 2010.

[8] Haenggi, M.; Andrews, J.G.; Baccelli, F.; Dousse, O.; Franceschetti, M., "Stochastic geometry and random graphs for the analysis and design of wireless networks," IEEE Journal on Selected Areas in Communications , vol.27, no.7, pp.1029,1046, September 2009.

[9] Pulido Herrera, Edith, et al. "Improving data fusion in personal positioning systems for outdoor environments," Information Fusion, vol. 14, no. 1, pp. 45-56, 2013.

[10] Bjelica, Milan Z., et al. "Cloud-enabled home automation gateway with the support for UPnP over IPv4/IPv6 and 6LoWPAN," IEEE International Conference on Consumer Electronics (ICCE), 2012.

[11] Blagodurov, Sergey, Sergey Zhuravlev, and Alexandra Fedorova. "Contention-aware scheduling on multicore systems." ACM Transactions on Computer Systems (TOCS), vol. 28, no.4, pp.8, 2010.

[12] M. Ghazanfari and G. M. Rezaei, An Introduction to Fuzzy Sets Theory, Iran University of Science and Technology Press, Tehran, Iran, 2006.

[13] Xie X, Wu P (2010) Research on the optimal combination of ACO parameters based on PSO. In: Proceedings of the international conference on networking and digital Society, vol 1, pp 94–97

[14] A. M. Ortiz, F. Royo, T. Olivares, J. C. Castillo, L. Orozco-Barbosa, and P. J.Marron, "Fuzzy-logic based routing for dense wireless sensor networks," Telecommunication Systems, vol. 52, no. 4, pp. 2687–2697, 2013.

[15] S. K. Sahani and K. Kumar, "Multi-routing in wireless sensor networks using an ant colony optimization (ACO)," International Journal of Computer Networking, Wireless and Mobile Communications, vol. 3, no. 3, pp. 87–98, 2013.

[16] J. Suhonen, M. Kuorilehto, M. H¨annik¨ainen, and T. D. H¨am¨al¨ainen, "Cost-aware dynamic routing protocol for wireless sensor networks—design and prototype experiments," in Proceedings of the IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06), pp. 1–5, Helsinki, Finland, September 2006.

[17] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using an ant colony optimization (ACO) router chip," Sensors, vol. 9, no. 2, pp. 909–921, 2009.

[18] B. Kadri, M. Feham, and A. Mhammed, "Efficient and secured ant routing algorithm for wireless sensor networks," International Journal of Network Security, vol. 16, no. 2, pp. 149–156, 2014.

[19] E. Ghazizadeh, M. Zamani, J.-L. A. Manan, and M. Alizadeh, "Trusted computing strengthens cloud authentication," The Scientific World Journal, vol. 2014, Article ID 260187, 17 pages, 2014.

[20] M. Alizadeh, W. H. Hassan, M. Zamani, S. Karamizadeh, and E. Ghazizadeh, "Implementation and evaluation of lightweight encryption algorithms suitable for RFID," Journal of Next Generation Information Technology, vol. 4, no. 1, pp. 65–77, 2013.

[21] D. Ma, J. Ma, L. Cheng, and P. Xu, "An adaptive virtual area partition clustering routing protocol using ant colony optimization for wireless sensor networks," in Advances in Wireless Sensor Networks, pp. 23–30, Springer, Berlin, Germany, 2014.