Open Access Library Journal
2024, Volume 11, e11567
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Research on Performance Improvement of Wireless Sensor Networks Based on OPM Algorithm

Pinggui Wu

College of Intelligent Science and Engineering, Chengdu Neusoft University, Chengdu, China
Email: 491718317@qq.com

How to cite this paper: Wu, P.G. (2024) Research on Performance Improvement of Wireless Sensor Networks Based on OPM Algorithm. *Open Access Library Journal*, **11**: e11567.
https://doi.org/10.4236/oalib.1111567

Received: April 12, 2024
Accepted: July 15, 2024
Published: July 18, 2024

Abstract

Traditional medical data collection methods are limited by equipment, space, time and other factors. Data transmission often has delay problems. We focus on wireless sensor network performance optimization OPM (Optimal routing scheme, Parallel computing, Maximum traffic and minimum overhead) algorithm. With CC2530 single-chip microcomputer as microcontroller and ZIGBEE as RF antenna, we build a wireless sensor network, obtain data from the serial port to transmit it to the cloud computing platform to form data set, adopt parallel computing technology, establish an optimal routing scheme, and deploy software with the algorithm with maximum traffic and minimum overhead to improve the data transmission speed of the wireless sensor network. Comparative experiments show that the data transmission speed is faster and the data security is higher. The OPM algorithm for wireless sensor network performance optimization is not only used to transmit medical data in wireless sensor networks, but also to other industry data.

Subject Areas

Computer Technology, Electronic Information, Communication

Keywords

CC2530, Software Performance, Wireless Sensor Networks

1. Introduction

(1) Research background of wireless sensor network performance

a) Promotion of technological development:

With the rapid advancement of wireless technology, wireless communication has developed to a relatively mature stage. The development of microelectromechanical systems and low-power, high-integration digital devices has enabled the realization of low-cost, low-power, and small-sized sensor nodes. These technological advancements provide a solid foundation for the development and application of wireless sensor networks (WSNs).

b) Wide application requirements:

Wireless sensor networks are widely used in various fields, such as battlefield surveillance, large-scale environmental monitoring, and target tracking. These application scenarios place high demands on the performance of wireless sensor networks, which has driven research on performance improvement.

c) Balanced requirements for low power consumption and high performance:

Due to the fact that wireless sensor network nodes are generally powered by batteries and often operate in harsh environments with a large number of nodes that are difficult to replace, low power consumption has become an important criterion for the design of wireless sensor networks. Achieving low power consumption while ensuring performance is an important direction of current research.

d) Challenges in data processing:

In wireless sensor networks, a large number of sensor nodes will generate massive amounts of data. How to effectively integrate and process these data, and improve data accuracy and transmission efficiency, is a key issue that needs to be addressed in performance improvement research.

e) Requirements for safety and reliability:

With the deepening of wireless sensor network applications, the issues of security and reliability have become increasingly prominent. In the future, wireless sensor networks need to adopt more advanced encryption and authentication technologies to improve network security and reliability.

f) Leading the future development trend:

Wireless sensor networks (WSN) are developing towards more intelligent, energy-saving, environmentally friendly and diverse directions. These trends have put higher demands on network performance and have prompted researchers to constantly explore methods and technologies for performance improvement.

The research background for improving the performance of wireless sensor networks is multifaceted, including the promotion of technological development, extensive application requirements, the balance between low power consumption and high performance, challenges in data processing, requirements for security and reliability, and the guidance of future development trends.

(2) Analyzing the previous research and summarizing the existing problems

The research on the performance of wireless sensor networks can summarize the following problems:

a) Energy limitation:

Wireless sensor network nodes usually carry limited energy sources, such as batteries, and these nodes are often deployed in inaccessible or dangerous environments, making energy replenishment difficult or impossible. Therefore, effective utilization and management of energy has become a key issue. Sensor nodes in the network may fail or become obsolete due to power depletion, which requires each node to minimize its own energy consumption during the operation of the wireless sensor network to achieve the longest working time.

b) Real-time requirements:

Many application scenarios of wireless sensor networks have high requirements for real-time performance. For example, in applications such as environmental monitoring or target tracking, sensor networks need to be able to respond quickly to events. If the network response is too slow, it may lead to the loss or invalidation of important data. Therefore, how to ensure the real-time performance of the network is a problem that needs to be addressed.

c) Data processing and fusion:

Due to the large number of nodes in wireless sensor networks, each node is constantly collecting data, resulting in a large amount of data in the network. These data need to be effectively fused and processed to reduce redundant information, improve data quality, and reduce transmission overhead. The purpose of data fusion technology is to fuse the data of multiple sensor nodes into useful information, reduce data transmission volume, and improve data accuracy. However, how to efficiently process these data remains a challenge.

d) Security and privacy protection:

The security of wireless sensor networks is one of the important factors that restrict their application. Due to the limited resources of sensor nodes and their deployment in unattended environments, they are vulnerable to various security threats. Researchers have proposed various encryption algorithms, intrusion detection mechanisms, and security protocols to enhance network security. At the same time, with the increasing awareness of privacy protection, how to protect user privacy while ensuring data availability has become a hot topic and challenge for research.

e) Cost issues:

Wireless sensor networks consist of a large number of sensor nodes, so the cost of a single node directly affects the cost of the entire network. In order to reduce system costs, it is necessary to design simple network systems and communication protocols that require low computing, communication, and storage capabilities. How to balance network performance while reducing node costs is a trade-off issue.

(3) This article addresses the energy constraints and real-time requirements in wireless sensor networks.

2. Optimal Routing Scheme

Using CC2530 [1] microcomputer as microcontroller and ZIGBEE as RF [2] antenna, a wireless sensor network is constructed, the data source generates data packets d at a constant speed s, the data flow continuous flow time is t, and the data length generated by the node is bit.

In the built wireless sensor network, the nodes are coordinator nodes, router nodes, and end device nodes. The coordinator node forms a wireless sensor network that can be responsible for receiving or transmitting signals; Router nodes are responsible for forwarding network information, which can be set to receive and transmit signals, often used in tree networks; The endpoint does not forward, only sends signals, is mostly asleep, and has low function consumption.

In the IEEE802.15.4 [3] standard, in order to control the number of sensor nodes described as joining the ZigBee network, four parameters need to be set. They are the maximum depth C_m of ZigBee wireless sensor network, the maximum number of descendant nodes that can be connected to the network source node L_m, the maximum number of descendant nodes to which the network source node can connect is C_m, the number of routing nodes that can be connected to the current node in ZigBee wireless sensor network R_m, the number of tree structure layers d, the offset C_{skip} for the network family is:

$$C_{skip}(d)=\begin{cases}1+C_m\times(L_m-d-1),\&R_m=1\\ \dfrac{1+C_m-R_m-C_m\times R_m^{L_m-d-1}}{1-R_m},\&otherwise\end{cases} \tag{1}$$

$$A_i=A_P+C_{skip}(d)\times(i-1)+1;(1\ll i\ll R_m) \tag{2}$$

$$A_k=A_P+C_{skip}(d)\times R_m+k;(1\ll k\ll C_m-R_m) \tag{3}$$

In the above formula, A_i is the address assigned by the coordinator or router node to the i-th routing node in ZigBee wireless sensor network, A_K is the address assigned to the k-th terminal node by the coordinator or routing node in the ZigBee wireless sensor network cable.

According to the ZigBee routing allocation principle, the tree routing algorithm in wireless sensor networks can only transmit upwards or downwards, although there are in the node data structure configuration table, but its advantages were not fully utilized during actual data transmission. Therefore, this article uses the node data structure configuration table for path selection to reduce average network latency and further improve the performance of the ZigBee network. The data structure configuration is as follows [4]: Identifier, Extension Address, Network Address, Node Type, Node Relationship, Remaining Energy.

Wireless sensors and networks [5] form a tree diagram including tree node structure, sleep state, location, family, data. The algorithm is represented in pseudocode, and its function is to find the optimal path algorithm. **Figure 1**: Optimal routing scheme algorithm is as follows.

In order to construct ZIGBEE [6] tree path optimization nodes, an empty queue is first created, and the tree nodes in the tree are sequentially added to the queue. Take the node with the minimum current path length as the current extension node, and sequentially check all vertices adjacent to the current extension node. If there is edge reachability from the current extension node i to vertex j, and the length of the corresponding path from the source through vertex i to vertex j is less than the current optimal path length, then the vertex is inserted as a live node into the live node priority queue. The expansion process of this node continues until the priority queue of the active node is empty.

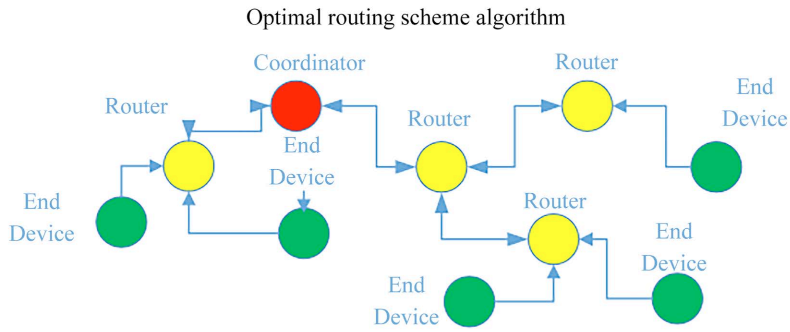

Figure 1. Optimal routing scheme algorithm.

The algorithm is as follows:

dist: An array of shortest distances, prev: Precursor vertex array, E: Current Extension Node, c: Adjacency matrix, H: Live node priority queue

while (true) {

for (int j = 1; j <= n; j++)

There is an edge between vertices i and j, and this path length is smaller than the original path length from the origin to j. This judgment achieves pruning

if ((c[E.node][j]<inf)&&(E.length+c[E.node][j]<dist[j])) {

// The vertex E.node is reachable to vertex j and satisfies control constraints

dist[j]=E.length+c[E.node][j];

prev[j]=E.node;

// Join the active node priority queue

MinHeapNode<Type> N;

N.node=j; // Vertex number is j

N.length=dist[j];

H.Insert(N);//Lack of upper bound function pruning

}

try {H.DeleteMin(E);}// Remove the next extension node

catch (OutOfBounds) {break;} //Priority queue empty

}

In order to construct ZIGBEE tree path optimization nodes, an empty queue is first created, and the tree nodes in the tree are sequentially added to the queue. Take the node with the minimum current path length as the current extension node, and sequentially check all vertices adjacent to the current extension node. If there is edge reachability from the current extension node *i* to vertex *j*, and the length of the corresponding path from the source through vertex *i* to vertex *j* is less than the current optimal path length, then the vertex is inserted as a live node into the live node priority queue. The expansion process of this node continues until the priority queue of the active node is empty.

The algorithm is as follows:

dist: An array of shortest distances, prev: Precursor vertex array, E: Current Extension Node, c: Adjacency matrix, H: Live node priority queue

```
while (true) {
        for (int j = 1; j <= n; j++)
There is an edge between vertices i and j, and this path length is smaller than the original path length from the origin to j. This judgment achieves pruning
                if ((c[E.node][j]<inf)&&(E.length+c[E.node][j]<dist[j])) {
// The vertex E.node is reachable to vertex j and satisfies control constraints
                    dist[j]=E.length+c[E.node][j];
                    prev[j]=E.node;
 // Join the active node priority queue
                    MinHeapNode<Type> N;
                    N.node=j;    // j Vertex number is j
                    N.length=dist[j];
                    H.Insert(N);//Lack of upper bound function pruning
          }
          try {H.DeleteMin(E);} // Remove the next extension node
          catch (OutOfBounds) {break;}    // Priority queue empty
}
```

The data obtained from the ZIGBEE [7] serial port is transmitted to the cloud platform to form a dataset, and the parallel computing technology is used to calculate the data set, and the calculation results are transmitted to the Web service interface to share the data for various application software system calls, Web service backend parallel computing uses a parallel computing method based on task allocation to break down tasks in the system into several subtasks (task1, task2, task3, …, task-n), and then enter the thread pool. Each subtask is assigned to each group (group1, group2, group3, …, group-n) by sequence number. Using the merge sorting method, obtain several minimum subtasks (MIN), call the operating system scheduler, and allocate them to the corresponding processor (CPU). As shown in **Figure 2**: Parallel computing model.

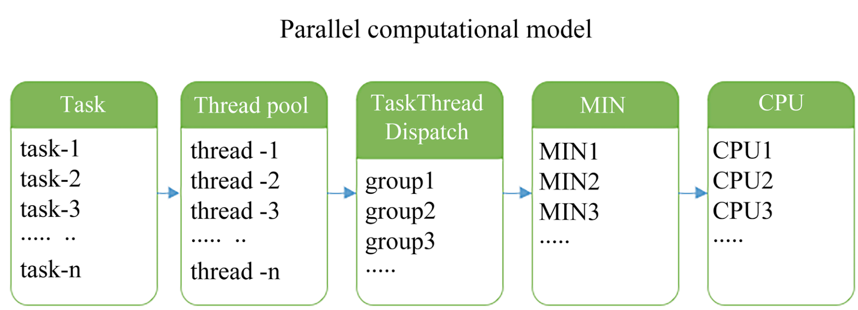

Figure 2. Parallel computational model.

The algorithm between the two nodes between the thread pool and the minimum subtask in **Figure 2** is as follows:

Select the first m smaller values from a sequence of length n.

(1) Input an unordered sequence A with a length of n = {a1, a2, a3, …, a_n};

(2) Divide the sequence into equal length groups, with the number of elements in each group greater than or equal to m;

(3) Output: The first m smallest in the sequence;

(4) Perform subtask division: Divide A into g = n/m groups, each containing m elements;

(5) Using the Batcher merge network to sort each group in parallel;

(6) Pairwise comparison: Compare each group of sorted sequences in pairs to form a MIN sequence;

(7) Repeat (5) and (6) for each MIN sequence until m smallest ones are finally selected;

(8) Assign the selected m smallest units to each CPU.

See Figure 3: Parallel computing subsystem diagram.

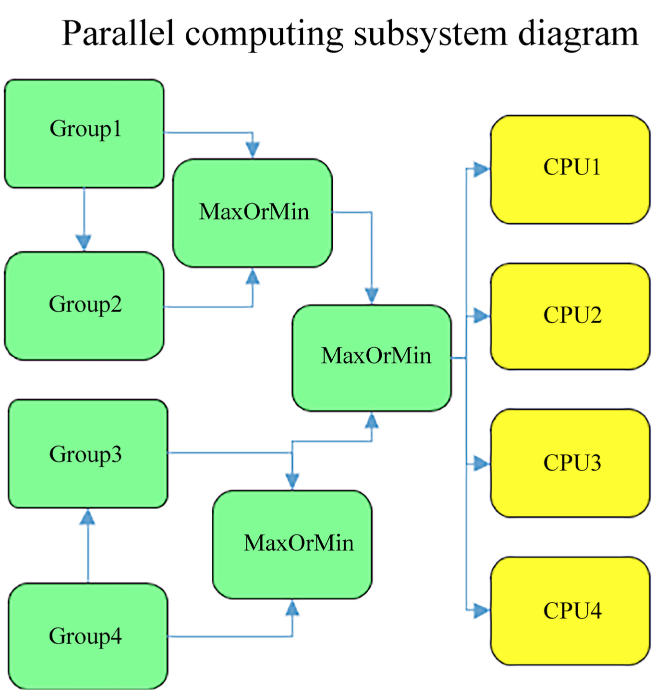

Figure 3. Parallel computing subsystem diagram.

Parallel computing uses multiple processors to reduce total processing time and does not use recursive function calls, resulting in good performance. The above time complexity analysis:

$$\begin{aligned} T\left(2^K\right) &= 2^{K-1}\left(1+2+3+\cdots+K\right) \\ &= 2^{K-1}\left(k\left(1+k\right)/2\right) \end{aligned}$$

n = 2^k, which is: $k = \log_2 n$

$$T(n) = 2^{\log^{n-1}} \log_2 \log_2 n\left(1+\log_2 \log_2 n\right)/2 = O\left(n.\log_2 n^2\right) \quad (4)$$

Through analysis, the time complexity is: $O\left(n.\log_2 n^2\right)$.

3. Maximum Traffic and Minimum Overhead

In order to optimize and improve the transmission performance of wireless sensor networks, software is deployed using algorithms with maximum traffic and minimum overhead. Define a server cluster as a directed graph [8] D = (V, A, C) where V is the set of server nodes in the graph, A is the set of network traffic, and C is the set of server node performance. V_S is the origin or source point, which only emits data; V_t is a receiving data node or sending data, also known as an intermediate point; C_{ij} is the transmission performance of the network line, b_{ij} is the unit traffic between two points in the network (v_i, v_j). The data flow on the network is a function $f = \{f_{ij}\} = f(v_i, v_j)$ on set A, f_{ij} is called the traffic on (v_i, v_j).

$$\min \sum_{(v_i, v_j) \in A} b_{ij} \tag{5}$$

For the intermediate point, outflow = injection, *i.e.* for each $i(i \neq s, t)$, there is:

$$\sum_{j:(v_i, v_j \in A)} f_{ij} = \sum_{k:(v_k, v_i \in A)} f_{ki} \tag{6}$$

For the starting point v_s. There are:

$$\sum_{j:(v_i, v_j \in A)} f_{sj} = v \tag{7}$$

For the ending point v_t. There are:

$$\sum_{k:(v_k, v_t \in A)} f_{kt} = v \tag{8}$$

When $D = v_{\max}$, it is the maximum traffic and minimum overhead [MTO].

Set the capacity of nodes vs to v2 to 5 and the flow rate to 3, denoted as ('vs','v2',5,3); The capacity of nodes vs to v3 is 3, and the flow rate is 6, denoted as ('vs','v3',3,6); The capacity of nodes v2 to v4 is 2, and the flow rate is 8, denoted as ('v2','v4',2,8); by analogy, they are recorded as: ('v3','v2',1,2), ('v3','v5',4,2), ('v4','v3',1,1), ('v4','v5',3,4), ('v4','vt',2,10), ('v5','vt',5,2).

The constructed network diagram is shown in **Figure 4**:

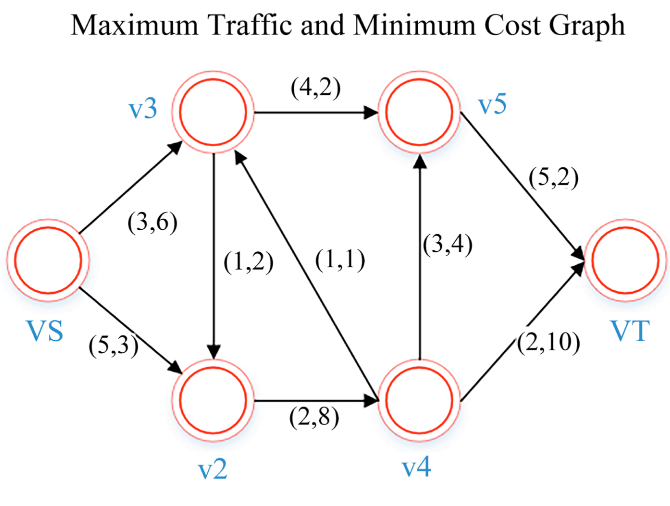

Figure 4. Maximum traffic and minimum cost graph.

The algorithm is as follows:

```
G=nx.DiGraph()#Constructing a directed graph without multiple edges
for k in range(len(L)):# L is the set of directed graph networks
    G.add_edge(L[k][0],L[k][1],capacity=L[k][2], eight=L[k][3])
maxFlow=nx.max_flow_min_cost(G,'vs','vt')
#Obtain the minimum cost and maximum flow
print("The maximum flow required is:",maxFlow)
mincost=nx.cost_of_flow(G, maxFlow) #The value of the minimum cost
print("The minimum server capacity is:", mincost)
node = list(G.nodes())    #Export Vertex List
n=len(node); flow_mat=np.zeros((n,n))
for i,adj in maxFlow.items():
    for j,f in adj.items():
        flow_mat[node.index(i),node.index(j)]=f
print("The maximum flow rate is:", sum(flow_mat[:,-1]))
print("The adjacency matrix of the minimum cost and maximum flow is:\n",flow_mat)
```

The above program code snippet can run the correct results.

4. Comparative Experiment

(1) Experimental purpose

Through comparative experiments, evaluate the differences in network performance between different wireless sensor network configurations, protocols, or optimization techniques, and provide a basis for selecting the best solution for practical applications.

(2) Experimental setup

Network topology: Build wireless sensor networks with star, tree, and Mesh topologies.

Node quantity: Deploy different numbers of nodes in each topology, such as 50, 100, and 200 nodes, to observe the impact of node quantity on network performance.

Transmission distance: Set different transmission distances to evaluate network coverage and communication quality.

Packet size: Send packets of different sizes to test the performance of the network under different loads.

Energy management strategy: Adopt different energy management strategies, such as sleep mechanism and power control, to evaluate their impact on network lifetime.

Routing protocol: Different routing protocols based on distance, energy awareness, or load balancing are used to compare their impact on network performance.

(3) Performance indicators

Network lifetime: Record the time from the network starting to run to the first node running out of energy.

Delay: Measure the average transmission time of data packets from the source node to the destination node.

Throughput: The amount of data successfully transmitted by the network in a unit time.

Packet loss rate: Statistics on the proportion of data packets lost during transmission.

Energy consumption: measure the total energy consumption of the network during operation.

(4) Experimental results and analysis

Through experiments, we can obtain the following data and analysis:

The impact of network topology: For example, with the same number of nodes, Mesh topology may have higher throughput and lower packet loss rate, but the latency may be relatively high.

The impact of node count: Increasing the number of nodes can usually improve network coverage and throughput, but it may also lead to higher energy consumption and latency.

The impact of transmission distance: An increase in transmission distance may lead to an increase in packet loss rate and a decrease in throughput.

The impact of packet size: Larger packets may lead to higher packet loss rates and latency, but can improve throughput.

The impact of energy management strategies: Adopting effective energy management strategies can significantly extend the network lifetime.

The impact of routing protocols: Different routing protocols have significant effects on network performance. For example, routing protocols based on the OPM Algorithm perform better in terms of network lifetime.

(5) Routing optimization algorithm ORS based on wireless sensor networks

i. Experimental equipment: 200 sensor nodes and 4 receiver computers.

ii. Experimental environment: Simulate a real hospital environment, with sensor nodes distributed in different locations. It is necessary to transmit patient life signals and other data to the receiving computer through a wireless sensor network. Use building a parallel computing algorithm model to read data.

iii. Test metric: The number of nodes in the wireless network has increased from 5 to 200.

Transmission time: 13 MS.

Data accuracy: Check whether the data received by the receiving computer is consistent with the data uploaded by the node.

(6) Experimental steps

i. Conduct medical data transmission experiments based on wireless sensor networks, recording transmission performance, data volume, and data accuracy;

ii. Conduct performance optimization algorithm experiments based on wireless sensor networks, recording performance, transmission data volume, and data accuracy;

iii. Compare and analyze the two technologies.

Compared with the MTO algorithm, the traditional ZIGBEE routing algorithm in wireless sensor networks starts from the source point and the total number of hops increases with the number of nodes. The traditional ZIGBEE routing scheme requires more hops and consumes longer energy and transmission time than the MTO algorithm. As shown in Figure 5:

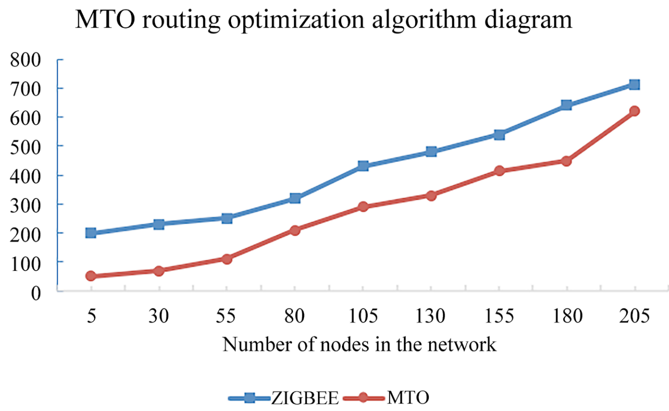

Figure 5. MTO routing optimization algorithm diagram.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Tahir, M., Javaid, N., Khan, M.A., Rehman, S., Javaid, A. and Khan, Z.A. (2014) Energy Efficient Transmission in Wireless Sensor Networks. *Research Journal of Applied Sciences, Engineering and Technology*, **7**, 723-727. https://doi.org/10.19026/rjaset.7.308

[2] Fabbri, F. and Buratti, C. (2011) Throughput Analysis of Wireless Sensor Networks via Evaluation of Connectivity and MAC Performance. In: Foerster, A. and Foerster, A., Eds., *Emerging Communications for Wireless Sensor Networks*. https://doi.org/10.5772/10514

[3] Khobahi, S., Soltanalian, M., Jiang, F. and Swindlehurst, A.L. (2020) Optimized Transmission for Parameter Estimation in Wireless Sensor Networks. *IEEE Transactions on Signal and Information Processing over Networks*, **6**, 35-47. https://doi.org/10.1109/tsipn.2019.2945631

[4] Al-Ghamdi, B., Ayaida, M. and Fouchal, H. (2019) Performance Evaluation of Scheduling Approaches for Wireless Sensor Networks. *Wireless Personal Communications*, **110**, 1089-1108. https://doi.org/10.1007/s11277-019-06775-3

[5] Liazid, H., Lehsaini, M. and Liazid, A. (2023) Data Transmission Reduction Using Prediction and Aggregation Techniques in IoT-Based Wireless Sensor Networks. *Journal of Network and Computer Applications*, **211**, Article ID: 103556. https://doi.org/10.1016/j.jnca.2022.103556

[6] Demirkol, I., Ersoy, C. and Alagoz, F. (2006) MAC Protocols for Wireless Sensor Networks: A Survey. *IEEE Communications Magazine*, **44**, 115-121. https://doi.org/10.1109/MCOM.2006.1632658

[7] Riaz, M.N., Qureshi, M.N. and Mahboob, D.A. (2011) Energy Efficient MAC Protocols for Wireless Sensor Networks: A Survey. *International Journal of Computer*

Science & Engineering Survey, **2** , 1859-1879.

[8] Yu, J.T., Yu, Z., Ding, M.L. and Ye, W. (2019) Research on the Tenacity Survivability of Wireless Sensor Networks. *Journal of Ambient Intelligence and Humanized Computing*, **11**, 3535-3544. https://doi.org/10.1007/s12652-019-01491-z