Journal of Information Systems and Telecommunication
Vol.11, No.2, April-June 2023, 123-134

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Cache Point Selection and Transmissions Reduction Using LSTM Neural Network

Maliheh Bahekmat[1], MohammadHossein Yaghmaee Moghaddam[1*]

[1]. Department of Computer Engineering, Ferdowsi University of Mashhad, Iran

Received: 12 Sep 2021 / Revised: 04 Jan 2022 / Accepted: 19 Feb 2022

Abstract

Reliability of data transmission in wireless sensor networks (WSN) is very important in the case of high lost packet rate due to link problems or buffer congestion. In this regard, mechanisms such as middle cache points and congestion control can improve the performance of the reliability of transmission protocols when the packet is lost. On the other hand, the issue of energy consumption in this type of networks has become an important parameter in their reliability. In this paper, considering the energy constraints in the sensor nodes and the direct relationship between energy consumption and the number of transmissions made by the nodes, the system tries to reduce the number of transmissions needed to send a packet from source to destination as much as possible by optimal selection of the cache points and packet caching. In order to select the best cache points, the information extracted from the network behavior analysis by deep learning algorithm has been used. In the training phase, long-short term memory (LSTM) capabilities as an example of recurrent neural network (RNN) deep learning networks to learn network conditions. The results show that the proposed method works better in examining the evaluation criteria of transmission costs, end-to-end delays, cache use and throughput.

Keywords: Reliability; Selection of Cache Points; Middle Caching; Wireless Sensor Networks.

1- Introduction

Wireless sensor networks also are used for the collection data for monitoring of environmental information. Reliability of data transmission in wireless sensor networks is very important in the case of a high lost packet rate due to link problems of buffer congestion. "Internet of Things" (IoT) is a modern technology in which any creature (human, animal, or object) can send data through communication networks, whether the Internet or intranet. The data sending process between IoT devices is automatic according to the configuration at specific times (usually permanently and instantaneously), without demanding the "human-to-human" or "human-to-computer" inter-action. Wireless Sensor Networks (WSNs) can play a significant role in promoting to cast the cheap and straightforward network for connecting IoT devices [5, 6]. However, battery-powered sensor nodes impose nodes to have a limited energy resource. Charging or replacing the sensor battery may be unpleasant or impossible in a work setting based on WSNs. Therefore, when the node loses its energy, it may not be efficient for assessment and monitoring [3, 4]. Therefore, one of IoT-based wireless sensor networks' critical problems is severe energy limitation [7]. Since these networks' efficiency depends a lot on the network's life span and network coverage, it is necessary to consider energy-saving algorithms to design IoT-based wireless sensor networks with long life.

Nowadays, researchers have developed dynamic management methods to overcome the energy consumption issues in IoT-based WSN's [8,9]. However, the increasing reliability rate is an essential concern for the dynamic management energy resources methods [10,11]. The reliability rate is relevant to the successful data transfer in IoT-based WSNs. In General, two mechanisms are used to be confident in IoT-based WSNs in practice: pack-based reliability and event-based reliability. Pack-based reliability requires sending all received data by sensor nodes to well, which could waste limited energy resources of nodes while event-based reliabilities do not need sending all received data, and it depends on sending the data covering the event. The areas related to sensor nodes are mostly interfering with each other, and for this reason, they are similar to each other in higher levels of sensed data. In monitoring applications,

🖂 **MohammadHossein Yaghmaee Moghaddam**
hyaghmae@um.ac.ir

the sensor network is in an environment that is supposed to monitor events such as fire or flood. This is possible with the application of small, cheap and smart sensor nodes. The sensors are equipped with low-power wireless interfaces used to communicate with each other. In environmental applications such as temperature and humidity monitoring, agricultural applications, urban life, 90 to 95% reliability is sufficient. But much more reliability is needed in military applications. Reliability means that all packets sent from the source must reach their destination and lost packets must be recovered by a secure schema. Reliability is calculated by the number of packets that reach the sink, not the reliability of individual packets. Reliability is directly related to energy efficiency. Sensor networks are a vital component of the Internet of Things (IoT) and are known as limited networks due to limited memory, computations, and energy capabilities.

One way to improve the reliability is to use local retransmission through interface caching. Data caching is an effective way for reducing the number of end-to-end re-transmissions, thereby reducing interference and overcoming variable channel conditions. Cache increases data access because it provides fast storage and retrieval of future information [1]. Caching techniques have a major impact on the transmission protocol proposed for WSN [2]. The development of the transport protocol should be independent of the other layers. A secure transport layer protocol is required in many wireless sensor network applications that provide different levels of reliability for different applications. Since congestion is one of the major causes of packet loss, congestion control mechanisms are a key component of the transport layer protocol. Congestion occurs when packets generated by sensor nodes exceed the network capacity. When congestion occurs in the network, the middle nodes destroy packets and this leads to the retransmission of packets and wasted energy in the network. Packet loss occurs not only due to congestion due to memory overload, but also for other reasons such as node mobility, node failure, collision, interference, and poor radio links. Hop-by-hop transmission is commonly used due to the short-range of sensor nodes in sensor networks. This increases the likelihood of packet loss and wasted energy to resend lost packets. To detect packet loss, explicit notification of lost packets has been proposed, which can be implemented as distributed (on sensor nodes) and centralized (on the sink). In the distributed model, sensor nodes use packet sequence numbers to identify lost packets. When a packet is lost, a middle node requests a re-transmission from its other neighbors. The sensor nodes detect the congestion from the buffer overflow. Therefore, to ensure the reliability of the network, the need for retrieval mechanisms such as retransmission and redundancy is felt.

Reliability can be divided into two levels [4]:

Packet or event confidence level

Hop-by-hop or end-to-end confidence level

data transmission in wireless sensor networks is in the case of a high lost packet rate due to link problems or buffer congestion. In this regard, mechanisms such as middle cache points and congestion control can improve the performance of the reliability of transmission protocols when the packet is lost. On the other hand, the issue of energy consumption in this type of networks has become an important parameter in their reliability

The contributions of this paper are as follows:

(1) Designing a reliable transport protocol using active cache management based on a deep learning algorithm and various cache management policies to improve cache performance.

(2) Providing a simulation and an analytical model to evaluate the performance of the cache-aware congestion control mechanism in the presence of lost packets in the WSN.

The rest of the paper is organized as follows: in section 2, the previously presented methods on sensor networks' reliability will be studied. Section 3 expresses the proposed method, section 4 evaluates the proposed method and studies its function, and finally, section 5 will give the overall conclusion.

2- Related Works

The data sending process between IoT devices is automatic according to the configuration at specific times (usually permanently and instantaneously), without demanding the "human-to-human" or "human-to-computer" inter-action. Wireless Sensor Networks (WSNs) can play a significant role in promoting to cast the cheap and straightforward network for connecting IoT devices

In [12], a new method has been presented for clustering wireless sensor nodes to reduce energy consumption named EAC. EAC is a clustering algorithm based on energy and distance; that means, sensor nodes are chosen as cluster head based on remaining energy. Meanwhile, non-cluster head nodes chose their cluster head based on distance from neighbor cluster heads. EAC algorithm increases the life span of network via balancing energy load among network nodes.

In [13], has presented a hierarchical clustering method for reducing energy consumption in wireless sensor network. This algorithm divides network to circles with different power levels in sinks and each circle has different nodes. Simulations and results obtained from using three scales of network life span, number of clusters and consuming energy of cluster heads showed that the efficiency of this method is better that LEACH in terms of energy consumption of cluster head, number of clusters and life span of network. In fact, this method reduces the number of dead node and energy consumption and increases the

life span of network. This algorithm includes three phases of launching, cluster launching and routing among clusters.
In [14], another hierarchical algorithm has been presented for reducing energy consumption in wireless sensor network. The presented method in this papers uses a mechanism to prioritize clusters and packs of data. This protocol provides a route without congestion for optimal energy consumption for necessary data packs via prioritizing. Therefore, the best route always remains for transferred vital and necessary data. Therefore, the proposed algorithm in this paper minimizes the delay and consuming energy and maximizes the life span of network and operational power.
BLAC algorithm [15] used combination of battery level and other criteria like node density and rank for choosing cluster head. For balancing in energy consumption, the cluster head is taken frequently by each node. In BLAC, the cluster heads gather data from his cluster sensor nodes and sends them via GPRS links.
In 2012, some researchers [16] studied the manner of forming clustering and especially a primary schema entitled fuzzy logic cluster formation protocol where fuzzy logic is used in clustering process. Several changes on several parameters related to fuzzy logic and clustering reduces the energy consumption and therefore increases the life span of network.
In [17], the optimization algorithm of single cluster network and multi-cluster networks are proposed where node energy harvesting are allocated to strengthening nodes chosen as cluster head and therefore, it leads to more survival of network. As it was said, the energy production resource in sensors is battery and this resource has less capacity. None of the above works could minimize the energy consumption and therefore the life span of wireless sensor networks will be less such that some of sensors will be eliminated after some time.
In [18], the neural networks were used for dynamic management of power (maximizing life span of sensor nodes after placement and for scheduling the cycle of the duties of sensor nodes (determining which node should be slept and which one should be alive). In this method, next event time is a non-fixed series which are predicted by wavelet neural network precisely. The mentioned neural network is a three-layer network which uses Morlt wavelet transform in hidden layer. The nodes which are in deeper sleeping modes consume less energy while it causes more delay and high energy consumption for waking.
In [19], a self-organizing neural network was used for reducing and classifying similar patterns. They used SOM in a hierarchical network architecture (based on the cluster) where nodes are an organization in several clusters and cluster head or sinks of data combinations. This self-organizing neural network reduces the transferring of data and classifies similar patterns.
Wireless sensor networks are widely used to perform the automations in many applications. The WSN is used in both attended and unattended environment such as Internet of Things, smart phones, health monitoring, surveillance, volcano monitoring, boarder surveillance and more The IoT based WSN are emerging rapidly because of its versatility and economic nature [20,22].
In many applications of wireless sensor networks, providing reliability and healthy delivery of packet to the destination is of great importance. Reliability is one of the tasks of the transport layer in these networks, which gives the network the ability to deliver data sent to the receiver securely [23.24].

3- The Proposed Method

Because WSNs are cost-effective and modular, they can be used to secure smart cities by providing remote monitoring and sensing for a variety of critical scenarios. In [1], a new framework for remote sensing and monitoring in smart cities using WSNs is proposed. In their proposal, they suggested using Unmanned Aerial Vehicles to act as a data mule to offload the sensor nodes and transfer monitoring data securely to the remote control center for further analysis and decision-making. Additionally, the paper provides insights into the implementation challenges of the proposed framework.
Machine learning technique is used in the proposed method of this paper to train the network based on various factors, such as buffer capacity, number of hops, energy node, speed of node, popularity and number of successful deliveries. The machine learning algorithm is trained based on previous network routing data to generate an equation, which examines the probability used, whether the communication node is able to deliver the message to the intended destination. To better understand the formula of this proposed method in Table 1 the notation of the formula is explained. This value is then used to decide on the next hop for the buffered message. In the training phase, long-term short-term memory capabilities are used to learn network conditions. The schema of this proposed method is shown in Figure 1.

Table 1: The notation of the formula

Notation	Description
P_f	Probability of failure due to failure
P_c	Probability of failure due to congestion
P_n	Probability of failure due to noise
P_{sd}	Probability of success to the destination
P_e	Bit loss probability
P_p	The probability of packet loss

PEP_i^f	Probability of progress of flow packets
$PER_{j,\, j+1}$	Probability of losing the packet on the link between nodes j and j + 1

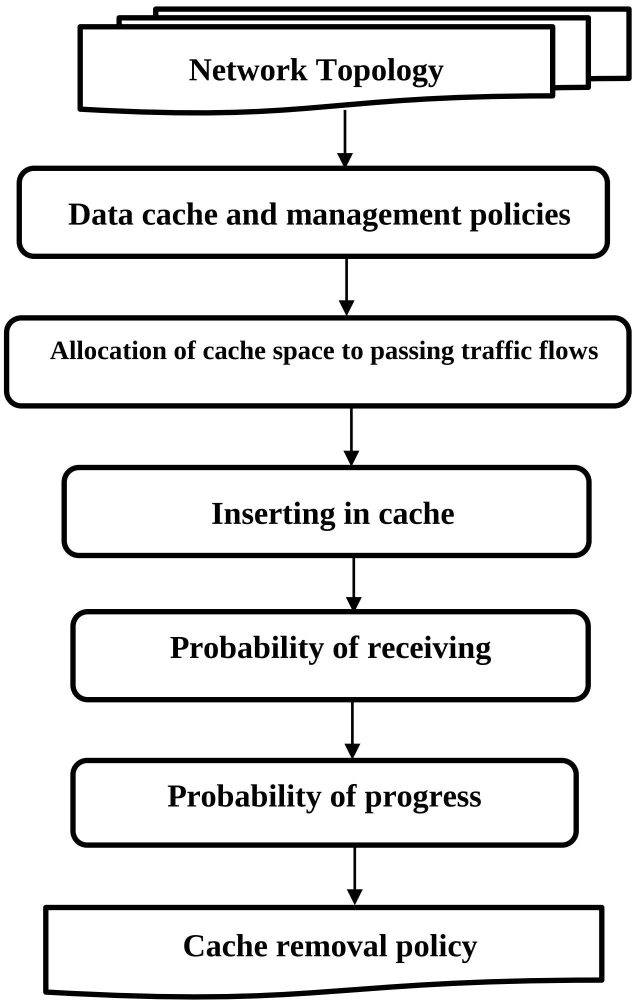

Fig. 1 Schema of the proposed method

- **Long Short Term Memory (LSTM) Cell**

LSTM is an artificial recurrent neural network (RNN) architecture [1] used in the field of deep learning. Unlike standard feed forward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as un-segmented, connected handwriting recognition speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

Table 2: LSTM with a forget gate [2]

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = a_g(W_f x_t + U_i h_{t-1} + b_i)$$

$$o_t = a_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \text{ o } c_{t-1} + i_t \text{ o } \tilde{c}_t$$

$$h_t = 0_t \text{ o } \sigma_h(c_t)$$

Variables

$x_t \in R^d$	: input vector to the LSTM unit
$f_t \in (0,1)^h$	: forget gates activation vector
$i_t \in (0,1)^h$	: input/output gates activation vector
$o_t \in (0,1)^h$	: output gates activation vector
$h_t \in (-1,1)^h$	: hidden state vector also know as output of the LSTM unit
$\tilde{c}_t(-1,1)^h$	: cell input activation vector
$c_t \in R^d$	: cell state vector
$W \in R^{h*d}, U \in R^{h*h}$ and $b \in R^h$	: weitht matrices and bias vector parameters

3-1- Input Parameters of the Training Phase

The proposed method uses machine learning techniques to select the next cache points. When a connection is established between two nodes and the buffer of one node contains a message to be transmitted, a decision must be made as to which node the message should be transmitted (as the next point selection). Normally, the message should only be sent from the sender to the adjacent recipient node, provided that the middle node has a sufficient probability of direct or indirect transmission to the destination node. Repeated message transmission can lead to packet loss and higher buffer overflow and higher power consumption. On the other hand, reducing frequent transmissions increases the number of unsuccessful delivery of messages. The probability of successful delivery depends on various factors that indicate the recent background and the ability of the nodes to deliver the message successfully. The probability of delivery in the next selected cache is calculated using a trained ML model, which includes the following features: delivery probability, buffer capacity, successful deliveries, success rate, source and destination node speed, distance from message source, distance to message destination, number of hops to current node, and message life time. The message life time parameter provides the duration from the creation of the message to the present. The message is transmitted from the transmitter node, if $P_m > P_r$ this point is considered as the cache point of this flow. In this expression, P_m is the probability of final delivery at the node and has been calculated using ML techniques. P_r is the current reliability rate at the transmitter node.

In the initial phase of the learning phase of this dissertation, the aim is to collect data related to network behavior analysis and then more effective items on network traffic management are selected by creating a database in the preprocessing phase. Then, in the second phase, the prediction accuracy increased by using the deep learning model based on long-short-term memory feedback neural networks (LSTM), we present a deep learning model so that we can improve the learning depth by deepening the time windows (short-term daily and long-term annual). In the third phase, the output of the deep model is given to the extreme learning machine (ELM), which can calculate the estimated delivery time of each packet instantaneously according to other input data. In the proposed method, we intend to use a stack of LSTMs due to the high sensitivity of time series data. The output of these networks is the probability of success to the destination.

- **Calculation of the Probability of Final Delivery Based on Machine Learning**

LSTM network: In the proposed method, the LSTM recursive neural network model with multiple hidden layers has been studied. Where (x1,…., X16) are the input parameters. Each training data sample provides input values (x1,…, x16) for selecting the next cache point, and the obtained output determines the probability of successful delivery of sent message to the destination. Prior to training, a sample network has been provided, that is, it has been quantified at random values and then the deep network is learned by repetition in the training set, thus it provides the desired predictions about whether the delivery will be successful or not. To calculate different p values, the machine learning model must first be prepared, trained, or constructed based on the training scenario data. Relevant data is called training data, and a specific data input is a training example. Then, during the next point selection process, the trained LSTM network is used to calculate P(Y) based on the input parameters (x16, ..., x1, x2), which are obtained in real time. The probability of successful sending by each node is denoted by P_{su} and is calculated by the following equation:

$$P_{\text{su}} = 1 - \mathrm{P}_{\text{loss}} \quad (1)$$

$$\mathrm{P}_{\text{loss}} = \mathrm{P}_{\text{f}} * \mathrm{P}_{\text{c}} * \mathrm{P}_{\text{n}} * \mathrm{P}_{\text{sd}} \quad (2)$$

Where, P_{f} is the probability of failure due to failure, P_{c} is the probability of failure due to congestion, P_{n} represents the probability of failure due to noise, and P_{sd} denotes the probability of success to the destination.

- **Calculation of the Probability of Packet Loss Due to Noise**

Assume that the wireless network has a bit loss probability equal to P_e. If the average length of packets is S bits, the probability of packet loss P_p can be calculated as follows:

$$p_p = 1 - (1 - p_e)^s \quad (3)$$

The probability of packet loss due to channel noise P_n is equal to P_p.

- **Calculation of the Probability of Packet Loss Due to Failure**

Another cause of packet loss in wireless sensor network nodes is hardware failures of the node. Assume that P_f represents the probability of loss due to hardware failure of the node. The failure rate of a sensor node depends on its external and internal factors. External factors affecting a node can be considered through the MIL-HDBK documentation. But internal factors need to be included in the final formula somehow. In the first step, the components of the system must be thoroughly examined in order to provide a series or parallel model or other modes that can be considered for it. The series mode for a system is the state in which the failure of each component of the system causes the failure of the entire system, and the parallel mode of a system is the state in which if all the components of a system fail, the entire system will fail. It is clear that the essential components of a node (microcontroller, memory, battery, communication device, sensor component, and ADC) form the system series; because the failure of each component causes the failure of the entire system. But the optional components, including the actuators and the number of sensors, form a parallel system. The probability of the proper function of a part or the reliability in the general case is obtained from the following equation.

$$R(t) = P_s = P(t_f \geq t \quad (4)$$

Given the above relation, the function f(t) must be specified. Given that the lifespan of most electronic components follows an exponential relationship, with the reliability of sensor node (R(t)), the probability of packet loss can be obtained as follows:

$$P_f = 1 - R(t) \quad (5)$$

- **Calculation of the Probability of Packet Loss Due to Congestion**

Each node has a buffer or queue. Its length and size can be a simple and good sign for congestion. The size of the buffer can be considered as a threshold. The proposed method uses a fixed threshold and if the buffer size exceeds the threshold, the congestion is detected. In some methods, the buffer size is periodically checked at the beginning of each period and the congestion is signaled instantaneously. The remaining length of the overall buffer size, or the difference between the remaining space and the traffic rate, can be used as a possible indicator of congestion. In the proposed method, by monitoring the average buffer queue length, a series of times is obtained that can use time delay neural network (TDNN) to predict its future values, which is actually the future status of the buffer of the cache point. For this purpose, we first design a suitable neural network and train it through the above time series. It then implements the trained neural network into the buffer point buffer to predict the future value of the average length of the buffer queue at a few time periods, called the forecast horizon, based on current and previous values of the average buffer queue length. Then, based on the result of this prediction, we introduce a mechanism called ML-RED, which operates on the basis of the RED algorithm, and the transmitters are notified before the congestion starts and reduce their transmission rate. One of the advantages of this method is that the determination of the cache points by the nodes is done locally and completely dynamically. This is done by comparing the transmission rate of each node with the required reliability interval for each stream. It should be noted that at the beginning of the network, before entering the learning stage, the cache points are randomly selected. In order to learn this, the cache points are selected or removed independently and locally by the middle nodes. Disclaimer of previous cache points only includes not inserting new entries and they are responsible for resending until deleted from the cache as a point cache. In the proposed method, congestion control is done globally and end-to-end and the selection of cache points is done locally by each node.

3-2- Data Cache and Management Policies

The PRM-DDCLAM (Deep PRM) protocol is a DTSN[1] protocol with sender side changes. This protocol uses both ACK and NACK messages that the recipient asks the sender to send through explicit acknowledgment request. The EAR signal is mounted on the data packet. After sending the EAR, the source launches the EAR timer. If the EAR timer expires before receiving the ACK/NACK, the source sends the EAR packet again. After adopting the EAR on the node of the receiver, a NACK, containing a bit map of the lost packets, is generated and sent to the sender. During the transport of such NACKs, cache points learn the lost packets and check if there are packets in the cache. If this is true, the cache points resend packets to the receiver and change the NACK bitmap before sending it to the sender. These implicit ACK and NACK notifications are a cache removal policy which are for removing all packets that have already been ACKed and also creating space for new incoming packets. Similarly, ML-PRM adapts a NACK repair mechanism so that cache points can output the NACK signal to speed up the repair process. In addition to receivers that can identify lost packets, cache points can also find lost packets and signal the node of the previous steps through the Repair Non- Acknowledgment Control Packet (RNACK), which includes a counter of lost packets. After receiving the RNACK, the node of the previous step, if it finds a copy in the cache, resends the lost packet to the destination. Otherwise, RNACK will be broadcast to the source. Assuming the route is always constant, RNACK transmission is not based on a timer, but occurs as soon as an out-of-row packet (outside the packet sequence) is detected. This capability further reduces the risk of packet loss by the accelerated recovery, thus it promotes instant transport.

3-3- Allocation of Cache Space to Passing Traffic Flows

In [16] and [17], the cache segmentation method has been used to manage the cache capacity. In this method, a cache point divides its cache capacity among the flows passing through the node in its route according to the cache segmentation policy in the network. To explain this method, suppose that ci represents the capacity of the node i and $\mathrm{F_i^n}$ is the total number of the flows passing through node i in its route. Using the cache segmentation policy, the weight $\mathrm{w_i^f}$ is assigned to each of these streams, which actually determines the part of the node cache i, which is assigned to flow f. By determining the weight of all the flows passing through node i, the true share of each flow from ci is determined by Equation (6):

$$\mathrm{p_i^f} = \frac{\mathrm{w_i^f}}{\sum_{\mathrm{j=1}}^{\mathrm{F_i^n}} \mathrm{w_i^j}} \qquad (6)$$

In this method, a packet of flow f can only be placed in the part of node i, which belongs to flow f, whose size is equal to $p_i^f \times c_i$

In the proposed method, after determining the optimal cache points of traffic flows passing through the network, considering the limited memory space of each middle node, the cache space of each node should be divided according to a specific policy between traffic flows passing through each node. The simplest cache segmentation policy is a uniform distribution policy in which the entire cache space is equally distributed between each traffic stream passing through it. This is certainly not a good policy. Our proposed solution in this section is to allocate the cache space of each node to the traffic flows passing through it based on various criteria such as: the requested reliability

[1] Distributed Transport for Sensor Networks

of the traffic flows and the distance of the node to the destination and the priority of the flow. In this way, traffic flows with high reliability and high distance of the nodes from the destination and have a higher flow priority, take up more space of the cache memory. The goal is to provide an exploratory algorithm based on effective parameters. To calculate the weighting coefficient of cache space allocation w_2^f, fuzzy systems with three inputs, the required reliability of traffic flows and the distance between the node and the destination and flow priority are used. As a result, the actual share of each flow from ci is determined by the following equation:

$$p_i^f = \frac{w_2^f w_i^f}{\sum_{j=1}^{F_i^n} w_i^j} \tag{7}$$

3-4- Inserting in Cache

Considering the limited resources in wireless sensor networks, it is crucial to provide designs that try to make optimal use of these resources. Transport layer protocols use cache in the middle nodes to counteract the inefficiency of the end-to-end retransmission method, enabling middle nodes to store packets received from different flows in their cache and resend them when needed. The purpose of storing packets in the middle nodes and locally retransmitting them is to minimize end-to-end retransmission, thereby reducing the number of transfers needed to send a packet from source to destination, energy consumption, and packet delay. To meet this goal, given the cache memory limitations, middle nodes must adopt appropriate policies to manage their cache space. Certainly one of the most important of these policies is the policy of selecting packets to be stored in the middle nodes.

A middle node decides to save a copy of the packet in its cache before sending it according to the packet selection policy. In an optimal packet selection policy, it should be tried that the packets with the experience of more difficult conditions in their downstream nodes as a result, lower probability of being received, have a higher chance of being stored in the middle node cache, so that they do not have to spend a lot of money to recover if they are lost along the route. On the other hand, a middle node in its packet selection policy should store packets with a lower probability of progress in the upstream nodes with a higher probability in its cache to prevent end-to-end retransmission. Undoubtedly, having a high cache capacity such that all incoming packets can be stored is the best option for middle nodes. However, given the limited resources (including memory) in sensor nodes, a middle node can only select a limited number of packets to store in its cache. On the other hand, providing a suitable policy for selecting packets depends on having a proper approach in determining the weight or in other words, the priority of each received packet in the middle nodes so that a suitable policy for selecting packets can be adopted through the weight of a packet. Therefore, in the following, we will examine the effective parameters in determining the weight of packets.

3-5- Effective Parameters

The purpose of examining the effective parameters in determining the weight of packets is to provide appropriate policies for selecting packets to be placed in the middle node cache. In such a way that these policies can serve the purpose of using cache in the middle nodes. To determine the effective parameters in packet selection, according to Figure (1), a middle node is considered as a destination for packets received from downstream nodes and, on the other hand, a source for the packets sent to the upstream nodes in the following route.

3-6- Probability of Receiving

In this section, we consider a middle node as the destination for packets received from downstream nodes. Therefore, we will examine the parameter affecting packet transport in the downstream nodes of a middle node. It should be noted that due to the nature of the message transmission in hop-by-hop form in a wireless sensor network, if a node sends a message to a destination in n farther hop, the probability of receiving the message by the destination is determined using Equation (8).

$$PoR_{\mathrm{n}} = \prod_{\mathrm{k=0}}^{\mathrm{n-1}}(1 - \mathrm{PER}_{\mathrm{k,k+1}}) \tag{8}$$

In an end-to-end retransmission scheme, if a message is lost along the route between sender and receiver, the source resends the message to the destination. In this case, the probability of receiving a message at least one time in the destination after sending for t times by the source is determined based on Equation (9):

$$\mathrm{r} = \sum_{\mathrm{k=0}}^{\mathrm{t-1}}(1 - \mathrm{PoR_n})^{\mathrm{k}}.\mathrm{PoR_n} \tag{9}$$

In Equation (2), $(1\text{- } P_oR_n)^k$ is the probability of not receiving a packet after k times of sending, and PoR_n is the probability of successfully receiving a packet after (k + 1) attempts. This equation shows that, after t times of sending a message by the source, it is expected that the message will be received with minimum reliability r at a destination with a distance of n hops. Lem (1) shows the geometric series introduced in Equation (10) as an equation of degree t.

$$\begin{aligned}
\mathrm{r} &= \sum_{k=0}^{t-1}(1 - PoR_n)^k.PoR_n \\
&= PoR_n.\sum_{k=0}^{t-1}(1 - PoR_n)^k \\
&= \frac{PoR_n.(1-(1-PoR_n)^t)}{1-(1-PoR_n)} \\
&= 1-(1-PoR_n)^t.
\end{aligned} \tag{10}$$

Equation (4), using Equation (3) obtained from Lemma (1), calculates the maximum number of transports required to achieve end-to-end reliability r:

$$\mathrm{r} = 1 - (1 - PoR_n)^t$$

$$1 - r = (1 - PoR_n)^t$$

$$t = \log_{(1-PoR_n)}(1 - r) \tag{11}$$

Equation (11) shows that in an end-to-end message transport design, a maximum of t (including retransmission) end-to-end transports by the source is required to provide a certain reliability (r) in delivering a message. As mentioned earlier, the use of cache in middle nodes tries to minimize end-to-end retransmission. Therefore, packet selection policies for caching in the middle nodes should select packets to be stored in the cache among the received packets, which will cause the greatest reduction in the number of end-to-end retransmissions (t).

3-7- Probability of Progress

The policy of selecting packets in a mid-node in addition to considering the conditions experienced by the received packets in downstream nodes, should also consider the status of packets in the route. In fact, a middle node, as the source of packets that it sends to upstream nodes, should try to store packets with low probability of placement in the cache of the upstream nodes or reaching their destination before being sent. In this section, we use the progress probability parameter to examine the condition of the received packets in the upstream nodes of a middle node. That is, the probability of storing packets of a flow in the cache of the upstream nodes of a middle node or reaching the destination. Equation (12) calculates the probability of progress of flow packets such as f in the upstream nodes of the middle node simplifies the description of Equation (12).

$$PEP_i^f = 1 - \sum_{j=i}^{l_f-1} \left[\prod_{k=i}^{j-1}\left(1 - PER_{k,k+1}\right).\left(1 - CP_k^f\right)\right].PER_{j.j+1} \tag{12}$$

In this equation $\left[\prod_{\mathrm{k=i}}^{\mathrm{j-1}}\left(1 - \mathrm{PER_{k,k+1}}\right).\left(1 - \mathrm{CP_k^f}\right)\right]$ represents the probability of reaching a packet of flow f is from node i to node j, in the case that none of the middle nodes between i and j are stored in the cache, and $\mathrm{PER_{j,\,j+1}}$ is the probability of losing the packet on the link between nodes j and j + 1. Node j can be any of the nodes upstream of node i (except destination).

Therefore;

$\left\{ \sum_{\mathrm{j=i}}^{\mathrm{l_f-1}} \left[\prod_{\mathrm{k=i}}^{\mathrm{j-1}}\left(1 - \mathrm{PER_{k,k+1}}\right).\left(1 - \mathrm{CP_k^f}\right)\right].\mathrm{PER_{j.j+1}} \right\}$ is the probability of non-progress of the flow packets f among all upstream nodes of node i. A packet selection policy used in middle nodes should store packets that have a low probability of progress in the middle nodes. Because these packets are likely to need to be resent and, the chances of upstream nodes being able to retrieve and resend these packets from their cache is low. On the other hand, storing packets in the middle node cache that have a high probability of progress will waste the cache capacity. Because these packets are most likely to be received by the destination or stored in the upstream middle nodes. In the first case, there is no need to resend the cached packets, and in the second case, if there is a need to resend, the middle nodes closer to the destination (due to receiving the lost message list sooner or faster timer expiration) will send the lost packets faster.

- **Local Variables and Packet Headers**

Implementing the proposed cache management system requires several local variables in the middle nodes, one field in the data packet header, and one field in the acknowledgment packet header.

To calculate the parameter of the probability of receiving each packet, we use the PoR field in the packet header. To explain the function of this method, let $\mathrm{PER_{i,\,i+1}}$ be the probability of lost packets on the link between nodes i and i + 1. Considering Equation (2), node i multiplies the value in the PoR field by (1- $\mathrm{PER_{i,i+1}}$) before sending the data packet to node i + 1 so that node i+1 is likely to receive this packet. Therefore, each middle node, given the PoR value of a packet, finds out the probability of receiving that packet again. The initial value of the PoR field is 1. To calculate the probability of progress of the packets of a flow, each node must store three local variables for each flow that passes through that node in its route. For example these variables for flow f are:

1. pep_flow (f): This variable specifies the probability of effective progress of the packets of flow f in the upstream nodes.
2. received_packet (f): This variable specifies the number of packets received from flow f.
3. cached_packet (f): This variable specifies the number of packets of flow f that have been successfully cached.

Calculating the effective progress probability parameter in addition to the variables mentioned will require a PEP field in the acknowledgement packet header. Given that the effective progress probability parameter for packets of a flow is calculated in the upstream nodes of a middle node. Therefore, we have mentioned the recursive form of Equation (12) in Equation (13) so that the acknowledgement packets that move in the opposite direction of the data packets from the destination to the source can calculate the probability of progress using the local variables of the middle nodes in the PEP field of their header. It should be noted that, like Equation (4-5), to calculate the effective progress probability, we use its opposite, that is, the probability of effective progress.

$$\mathrm{PEP_i^f} = 1 - \left[\mathrm{PER_{i,i+1}} + \left(1 - \mathrm{PER_{i,i+1}}\right)\mathrm{NPEP_{i+1}^f}\right] \tag{13}$$

$$NPEP_i^f = \left(1 - PEP_i^f\right)\left(1 - CP_i^f\right) \tag{14}$$

$$NPEP_h^f = 0 \tag{15}$$

PEP_i^f represents the probability of progress of the packets of flow f among the upstream nodes of node i and $NPEP_i^f$ is the probability that the packets of flow f will not progress among the upstream nodes of node i and will not be stored on the cache of node i. Specifically, the probability of non-progress of the packets of a flow in the upstream nodes of the destination (h) and non-caching on the destination node is zero. In each node, as the acknowledgment message is received and after calculating PEP_i^f, this value is placed in the local variable pep_flow (f) corresponding to the flow f, then $NPEP_i^f$ is calculated and an acknowledgment message is placed in the PEP field and sent to the next node in the direction of source. Therefore, after receiving an acknowledgment message from a flow such as f in a middle node such as i, the value $NPEP_{i+1}^f$ will be in its PEP field, and this value will be changed to $NPEP_i^f$ before sending for node i-1. Each middle node to calculate PEP_i^f requires the calculation of CP_i^f, which is obtained based on Equation (16):

$$CP_i^f = cached(f)/recived(f) \tag{16}$$

In this method, each middle node updates the probability of the progress of the packets by receiving an acknowledgment message from a flow such as f. The packet selection policy in the proposed cache management system, in addition to the parameters of probability of receipt and effective progress, also requires the parameter distance from the source. To calculate this parameter, we use the ttl field in the data packet header.

3-8- Cache Removal Policy

Any packet stored in the cache of a middle node will be deleted from the cache only by replacing or receiving an acknowledgment message. But if the cache does not have enough space to store the new packet and the input packet weighs more than the packet in the cache can be selected from the packets stored in the cache of the packet whose remaining lifespan is ending to be replaced.

4- Experimental Results

In this section, we will evaluate the proposed cache management system. The network topology intended for evaluation is a 10×10 grid topology, which is shown in the figure 2. In this topology, each of the nodes at the end of the communication links sends a flow of data to the corresponding nodes on the opposite side. Hence each of these nodes sends a data flow and receives a data flow. This approach allows four data flows to pass through each middle node. The direction of these flows and the number of flows passing through the middle nodes have been specified in the figure 2.

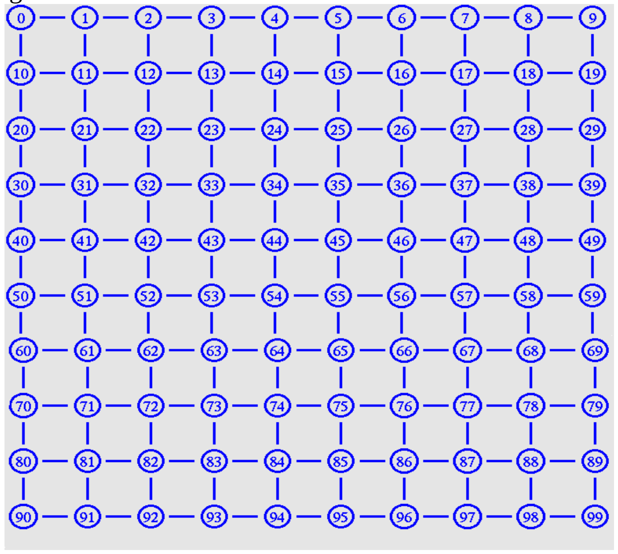

Fig. 2 10 × 10 grid topology for simulation

The simulation was used by ns-2 simulator to evaluate the proposed cache management system. This simulator, which is based on a discrete and object-oriented event simulation method, was designed and implemented by Berkeley University and, it is one of the best tools available to researchers to simulate wired and wireless networks.

In the scenario considered for the simulation, the source of each flow sends 250 packets to the destination, which due to the presence of 40 flows, a total of 10000 packets have been sent in each time of the simulation run in this network. The start time of each flow was randomly ranged from 1 to 1 second and the data transmission rate is fixed. The results of the evaluations have been presented for the proposed method and the comparative method presented in [24] as the Based Method.Two different scenarios have been used to compare the results of the two methods. In the first scenario, the error rate between nodes is considered between 0.2 and 0.6 with intervals of 0.05, while the cache size of the middle nodes has been considered to be 25. But in the second scenario, the error rate between the links is fixed and between 0.2 and 0.3, but the cache size of the middle nodes is between 5 to 40 with intervals of 5 units. In both scenarios, the goal was to achieve reliability between 0.8 and 0.9. The results of these two scenarios have been presented in the form of evaluation.

4-1- Transmission Cost

Given the energy constraints on sensor nodes and the direct relationship between energy consumption and the number of transmissions made by the nodes, a cache management system should try to minimize the number of transmissions needed to send a packet from source to destination. We define the transmission cost as the average number of transmissions required to send a packet from source to destination. In this subsection, we will examine the transmission cost in the proposed methods. Figure 3 and Figure 4 show the transmission cost in comparative methods for different error rates and cache sizes.

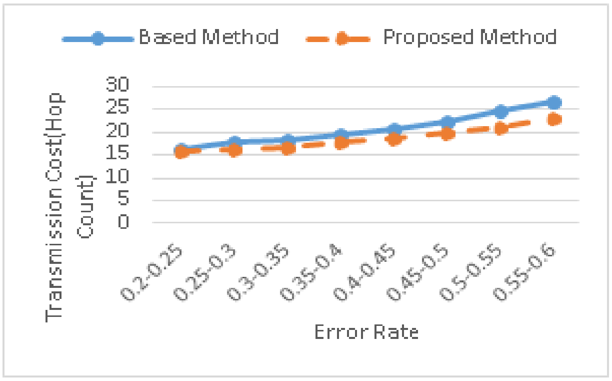

Fig. 3 Investigation of transmission costs in the proposed method and the based method with different error rates

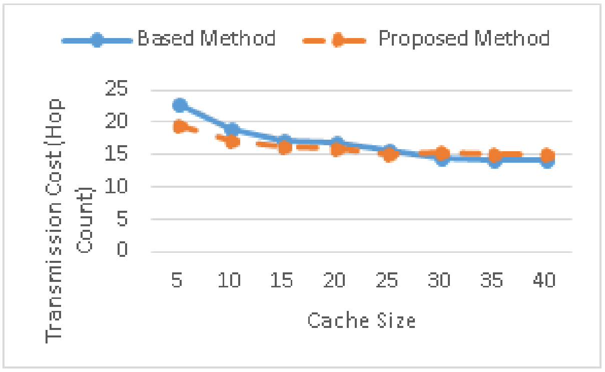

Fig. 4 Investigation of transmission costs in the proposed method and the based method with different cache sizes

4-2- End-to-end Delay

The delay calculated in this section is based on the definition of end-to-end delay. End-to-end delay is the time between the first sending of a packet and the successful receipt of the packet sent to the destination. Figure 5 and Figure 6 illustrates the end-to-end delay.

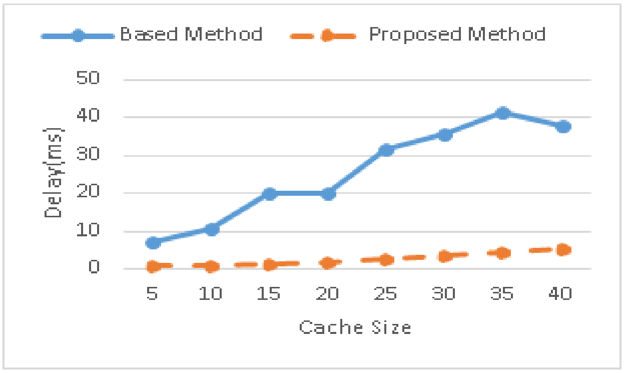

Fig. 5 Delay of sent packets at different error rates

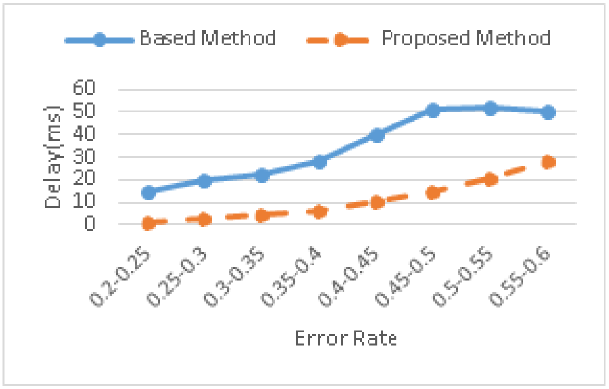

Fig. 6 Delay of sent packets with different sizes

4-3- Cache Use

Figure 7 and Figure 8 show the cache use in the middle nodes for different scenarios in the proposed methods.

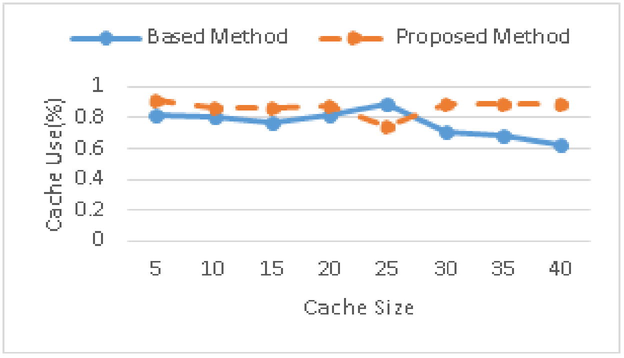

Fig. 7 Cache use with different error rates

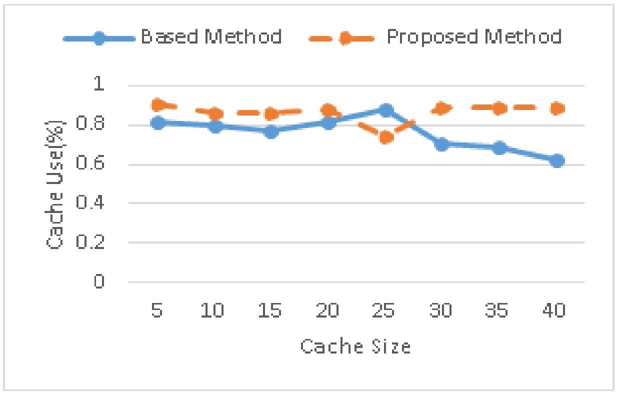

Fig. 8 Cache use with different cache sizes

4-4- Throughput

Throughput as another service quality parameter, plays an important role in determining the quality of the compared methods. Therefore, in this section, the throughput is examined. In Figure 9 and Figure 10, the throughput of each method is reported for different scenarios.

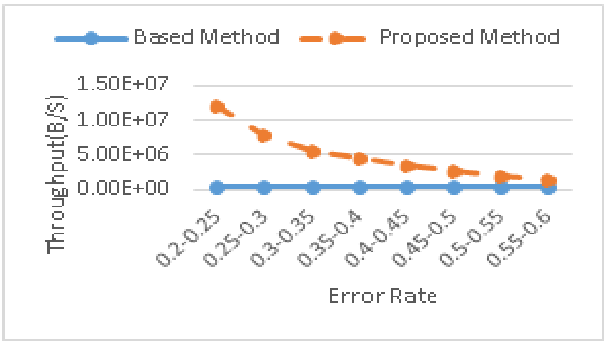

Fig. 9 Throughput with different error rates

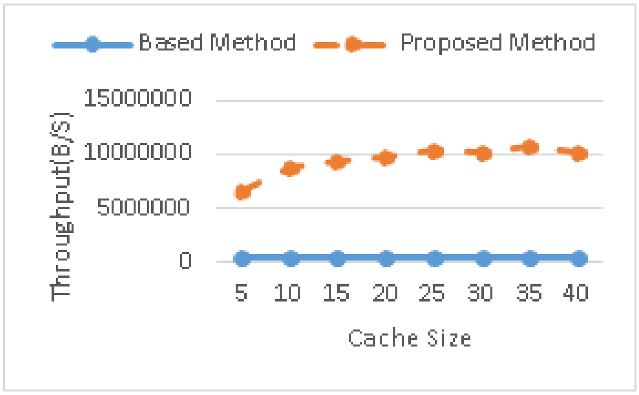

Fig. 10 Throughput with different cache sizes

5- Conclusion and Future Works

Wireless sensor networks are widely used to perform automation in many applications. The WSN is used in both attended and unattended environments such as the Internet of Things, smartphones, health monitoring, surveillance, volcano monitoring, border surveillance, and more The IoT-based WSN is emerging rapidly because of its versatility and economic nature.In many applications of wireless sensor networks, providing reliability and healthy delivery of the packet to the destination is of great importance. Reliability is one of the tasks of the transport layer in these networks, which gives the network the ability to deliver data sent to the receiver securely. In this paper, a protocol for providing statistical reliability for multiple traffic flows using dynamic caching capability is presented. In order to make optimal use of memory and energy, distributed dynamic caching methods have been considered. On the other hand, in many applications of wireless sensor networks, providing 100% reliability is not considered and statistical reliability is also required. In this dissertation, a new protocol for recovering lost packets in these networks has been presented, which has the following features:

• Suitable for applications that need to provide statistical reliability. Naturally, this protocol can also be used for applications with 100% reliability.

• Ability to support various flows with different characteristics (different packet lengths).

• Act in caching mode to balance energy consumption and memory.

• In calculating the probability of loss, all the factors that produce loss, including channel noise, node failure, and congestion, are considered as the input of the deep learning network.

• Caching points are dynamic and are dynamically determined according to the needs of traffic flows and network conditions.

• Has the ability to support different traffic classes.

• Can be extended to heterogeneous wireless sensor networks. Implementations of different scenarios in the results show that the proposed method works better in examining the evaluation criteria of transmission costs, end-to-end delays, cache use, and throughput.

References

[1]. A. Khalifeh et al., “Wireless sensor networks for smart cities: Network design, implementation and performance evaluation, Electron” vol. 10, no. 2, pp. 1–28, 2021, doi: 10.3390/electronics10020218.

[2]. A. Shankar, N. R. Sivakumar, M. Sivaram, A. Ambikapathy, T. K. Nguyen, and V. Dhasarathan, “Increasing fault tolerance ability and network lifetime with clustered pollination in wireless sensor networks,” J. Ambient Intell. Humaniz. Comput., vol. 12, no. 2, pp. 2285–2298, 2021, doi: 10.1007/s12652-020-02325-z.

[3]. Kabashkin, I., Reliability of Cluster-Based Nodes in Wireless Sensor Networks of Cyber Physical Systems. Procedia Computer Science, 2019. 151: p. 313-320.

[4]. W. Osamy, A. A. El-Sawy, and A. Salim, "CSOCA: Chicken Swarm Optimization Based Clustering Algorithm for Wireless Sensor Networks," IEEE Access, vol. 8, pp. 60676–60688, 2020, doi: 10.1109/ACCESS.2020.2983483.

[5]. S. K. S. L. Preeth, R. Dhanalakshmi, and P. M. Shakeel, "An intelligent approach for energy efficient trajectory design for mobile sink based IoT supported wireless sensor networks," Peer-to-Peer Netw. Appl., vol. 13, no. 6, pp. 2011–2022, 2020, doi: 10.1007/s12083-019-00798-0.

[6]. Zhou, Y., et al., Topology design and cross-layer optimization for wireless body sensor networks. Ad Hoc Networks, 2017. 59: p. 48-62.

[7]. R. Arya and S. C. Sharma, "Analysis and optimization of energy of Sensor node using ACO in wireless sensor network," Procedia Comput. Sci., vol. 45, no. C, pp. 681–686, 2015, doi: 10.1016/j.procs.2015.03.132.

[8]. Singh, A., Sharma, S. and Singh, J., 2021. Nature-inspired algorithms for wireless sensor networks: A comprehensive survey. Computer Science Review, 39, p.100342.

[9]. A. Singh, S. Sharma, and J. Singh, "Nature-inspired algorithms for Wireless Sensor Networks: A comprehensive survey," Comput. Sci. Rev., vol. 39, 2021, doi: 10.1016/j.cosrev.2020.100342.

[10]. H. Zhong, L. Shao, J. Cui, and Y. Xu, "An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks," J. Parallel Distrib. Comput., vol. 111, pp. 1–12, 2018, doi: 10.1016/j.jpdc.2017.06.019.

[11]. N. Abbas and F. Yu, "Design and Implementation of a Video Surveillance System for Linear Wireless Multimedia Sensor Networks," 2018 3rd IEEE Int. Conf. Image, Vis. Comput. ICIVC 2018, pp. 524–527, 2018, doi: 10.1109/ICIVC.2018.8492776.

[12]. M. Mohamed-Lamine, "New clustering scheme for wireless sensor networks," 2013 8th Int. Work. Syst. Signal Process. Their Appl. WoSSPA 2013, pp. 487–491, 2013, doi: 10.1109/WoSSPA.2013.6602412.

[13]. Meenakshi, D. and Kumar, S., 2012. "Energy Efficient Hierarchical Clustering Routing Protocol for Wireless Sensor Networks". Springer, pp. 409-420.

[14]. N. Goel and G. Aujl, "Simulation and feasibility analysis: Hierarchical Energy Efficient Routing Protocol (HEERP) for Wireless Sensor Network," Int. Conf. Commun. Signal Process. ICCSP 2013 - Proc., pp. 1143–1148, 2013, doi: 10.1109/iccsp.2013.6577235.

[15]. T. Ducrocq, N. Mitton, and M. Hauspie, "Energy-based clustering for wireless sensor network lifetime optimization," IEEE Wirel. Commun. Netw. Conf. WCNC, pp. 968–973, 2013, doi: 10.1109/WCNC.2013.6554695.

[16]. R. Mhemed, N. Aslam, W. Phillips, and F. Comeau, "An energy efficient fuzzy logic cluster formation protocol in wireless sensor networks," Procedia Comput. Sci., vol. 10, pp. 255–262, 2012, doi: 10.1016/j.procs.2012.06.035.

[17]. P. Zhang, G. Xiao, and H. P. Tan, "Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors," Comput. Networks, vol. 57, no. 14, pp. 2689–2704, 2013, doi: 10.1016/j.comnet.2013.06.003.

[18]. Y. Shen and X. Li, "Wavelet neural network approach for dynamic power management in wireless sensor networks," Proc. Int. Conf. Embed. Softw. Syst. ICESS 2008, pp. 376–381, 2008, doi: 10.1109/ICESS.2008.36.

[19]. Oldewurtel F, Mahonen P. Neural wireless sensor networks. In2006 International Conference on Systems and Networks Communications (ICSNC'06) 2006 Oct 6 (pp. 28-28). IEEE.

[20]. Chou LD, Li DC, Hong WY. Improving energy- efficient communications with a battery lifetime- aware mechanism in IEEE802. 16e wireless networks. Concurrency and computation: Practice and experience. 2013 Jan;25(1):94-111.

[21]. P. Mohanty and M. R. Kabat, "A hierarchical energy efficient reliable transport protocol for wireless sensor networks," Ain Shams Eng. J., vol. 5, no. 4, pp. 1141–1155, 2014, doi: 10.1016/j.asej.2014.05.009.

[22]. B. Sharma and T. C. Aseri, "A hybrid and dynamic reliable transport protocol for wireless sensor networks," Comput. Electr. Eng., vol. 48, pp. 298–311, 2015, doi: 10.1016/j.compeleceng.2015.01.007.

[23]. Tiglao, N.M.C. and A.M. Grilo. Transmission Window Optimization for Caching-Based Transport Protocols in Wireless Sensor Networks. 2015. Cham: Springer International Publishing.

[24]. Alipio MI, Tiglao NM. RT-CaCC: a reliable transport with cache-aware congestion control protocol in wireless sensor networks. IEEE Transactions on Wireless Communications. 2018 Apr 24;17(7):4607-19.