Intelligent and Converged Networks ISSN 2708-6240
2024, 5(1): 68−80 DOI: 10.23919/ICN.2024.0006

Fuzzy and IRLNC-based routing approach to improve data storage and system reliability in IoT

U. Indumathi* and A. R. Arunachalam

Abstract: Internet of Things (IoT) based sensor network is largely utilized in various field for transmitting huge amount of data due to their ease and cheaper installation. While performing this entire process, there is a high possibility for data corruption in the mid of transmission. On the other hand, the network performance is also affected due to various attacks. To address these issues, an efficient algorithm that jointly offers improved data storage and reliable routing is proposed. Initially, after the deployment of sensor nodes, the election of the storage node is achieved based on a fuzzy expert system. Improved Random Linear Network Coding (IRLNC) is used to create an encoded packet. This encoded packet from the source and neighboring nodes is transmitted to the storage node. Finally, to transmit the encoded packet from the storage node to the destination shortest path is found using the Destination Sequenced Distance Vector (DSDV) algorithm. Experimental analysis of the proposed work is carried out by evaluating some of the statistical metrics. Average residual energy, packet delivery ratio, compression ratio and storage time achieved for the proposed work are 8.8%, 0.92%, 0.82%, and 69 s. Based on this analysis, it is revealed that better data storage system and system reliability is attained using this proposed work.

Key words: Internet of Things (IoT); data storage management; fuzzy system; improved random linear network coding; energy utilization; system reliability

1 Introduction

Internet of Things (IoT) helps in integrating physical gadgets with the internet, and it is specifically a network infrastructure which consists of a huge number of tiny sensors for communication[1]. The major function of this deployed sensor is to sense the information from the deployed environment and transmit it to sever for visualization. Due to its tiny nature and cheaper installation cost, it is widely utilized in various applications such as agriculture, smart building, smart city, industry, emergency areas and so on for performing numerous operations[2]. However, in case of an emergency situation, the sensor nodes are deployed in a hostile environment, so they are frequently prone to failure and this result in decreased data transmission and system reliability. Therefore to improve network performance, the researchers are focusing on developing distributed data storage management system, which also helps in maintaining data redundancy[3, 4].

Data packets are replicated in several physical storage devices in a standard data storage system and in the event of device failure due to severe conditions, data is retrieved from the surviving nodes[5]. Data replication, on the other hand, is simple, but it is usually small in terms of storage and energy use. Induced data redundancy in each node, on the other hand, is possible[6]. However, the resulting data redundancy leads to increased data traffic, which may be too much for energy-constrained sensor networks to handle[7]. In addition, controlling the growing volume of data causes significant overhead in the storage nodes. When constructing data storage strategies, the cost of storage should also be considered[8]. Therefore,

• U. Indumathi and A. R. Arunachalam are with the Department of Computer Science and Engineering, Dr. M.G.R. Educational and Research Institute, Chennai 600095, India. E-mail: indumathi.mgr@gmail.com; abkakade22@gmail.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2022-12-09; revised: 2023-07-04; accepted: 2023-08-20

to improve data redundancy and reduce data traffic and energy consumption, an efficient algorithm must be designed.

In addition to storage problems, security is also a major concern in sensor networks because it is frequently prone to various attacks and threats[9]. To improve system reliability, the utilization of network coding techniques is a recent trend. Network Coding (NC) seems to be an encoding/decoding technology that aims to improve overall network performance while reducing delays. Intermediate nodes use this approach to process and encrypt received packets before resending them[10]. In communication networks, there are three types of coding mechanisms: source coding, which uses compression to reduce data redundancy and resources, channel coding, which uses redundancy to improve reliability across a lost media, and network coding, which sits in between these two levels[11, 12]. So, to improve overall network performance, an efficient algorithm must be designed to overcome both storage and reliability problems. Moreover, the existing work focus either on storage problem or system reliability.

The major contribution of this work is discussed below.

Efficient routing algorithm based on fuzzy and Improved Random Linear Network Coding (IRLNC) with DSDV is designed to improve fault tolerance mechanism in IoT based sensor networks due to the deployment of sensor nodes in harsh environment.

Management of data is improved by using the fuzzy system, which is based on generated fuzzy rules with the workload, residual energy and coding capability for selecting the optimal storage nodes.

For compression and encoding the data packets transmitted from source node, the IRLNC is used for efficient storage management and securing the data from various attacks. This model is similar to the existing Random Linear Network Coding (RLNC) method but the compression of the data is an add-on contribution for the designed model, which results in improved system reliability such as packet delivery ratio and secure data with lossless transmission.

Transmitting of encoded data from the storage node to destination node with the shorted path is performed by using the Destination Sequenced Distance Vector (DSDV) algorithm for reducing the energy consumption.

The remaining section of the paper is organized as follows. Section 2 comprises articles related to data storage management and network coding. Section 3 covers a detailed description of the proposed system in IoT platform. Section 4 comprises the results obtained through performing simulation analysis. Finally, section 5 concludes the entire research work.

2 Literature review

Storage node deployment and data routing are found to be the most challenging task in sensor networks. In addition to it, maintaining the security of the network is also found to be a complex task and it is achieved by means of various network coding techniques. Some of the data storage management based routing algorithm and network coding technique is reviewed below.

2.1 Network coding

The network coding backpressure routing algorithm was developed by Malathy et al.[13] to optimize power consumption in IoT networks. The main intention of applying the backpressure algorithm was to divert the flow of packets. In this work, the battery power was optimized by means of flowing the packet from a heavily congested node to a low congested node. The lossy compression algorithm that functions on the basis of Bayesian predictive coding was designed by Chen et al.[14] This predictive coding technique sends the error term rather than the original signal to the intermediate node. Error terms were obtained by subtracting the predicted signal from that of the actual signal. The compression functioning of this developed approach depends on the prediction approach. In this developed approach, the Bayesian interference was combined along with predictive coding. Network coding embedded braided multipath routing approach was designed by Li et al.[15] for reducing transmission delay and energy consumption in a network system. Major issue faced in network system was decreased rate of data transmission and system reliability. This issue was mainly due to the failure of node and link loss. For that purpose network coding based routing protocol was designed. This developed network coding was integrated along with compression and Hierarchical Multi-Parent Nodes (HMPNs) architecture for routing. Tracey Ho et al.[16] had developed a method of multicasting from many sources over a network using randomized network coding, in which nodes separately and arbitrarily choose the linear mappings of inputs onto output connections over a field. From this we obtain a success probability bound for randomized network coding in link-redundant networks with unreliable links, in terms of link failure probability and amount of redundancy.

Ho et al.[17] had presented a novel randomized network coding approach for robust, distributed transmission and compression of information in networks. In this techniques, the storage and energy management for the data is not effective due to the large size and long path of data transmission from the nodes. Thus, the proposed model uses an efficient IRLNC technique for compressing the large data without any information loss as well as encoding of the data.

2.2 Data storage management

An effective and reliable data storage approach was introduced by Chervyakov et al.[18] to improve security in IoT by means of detecting and correcting errors. This developed approach utilized approximate values related to the rank of a number to overcome the complexity faced during the decoding process. Additionally, arithmetic properties of the residue number system were also used for detection and correction of error. Theoretical analysis was performed to study decoding and encoding time, data redundancy and information loss. An effective algorithm was developed by Yang et al.[19] to improve both storage node installation and optimal routing selection. Data storage and system reliability were found to be serious problems in case of heterogeneous networks. So, in this developed approach, data redundancy and effective deployment of storage nodes were achieved using the coding technique. However, these existing techniques store a huge amount of data which results in increased storage cost and energy consumption in nodes. At the same time, only limited research is reported on optimizing both data storage and system reliability. To overcome these problems, an efficient algorithm based on IRLCN and DSDV is designed that combines both data storage and system reliability.

3 Proposed system in IoT

IoT based wireless sensor networks are employed in various day-to-day applications for transmitting a vast amount of useful information. It can be used in various application including health care, transportation, etc. Normally, the information gathered through source and intermediate nodes is transmitted to the sink node for storage. During this entire process, there is a possibility of data corruption in the middle of the transmission, which ultimately leads to huge data loss. To address this problem, the concept of data storage management is introduced for retaining sensed information. In this system, the intermediate nodes act as storage nodes and the data acquired through the sensor nodes are delivered to the storage node by means of multihop transmission. Moreover, these storage nodes are located in hostile environments such as earthquake regions and battlefields. Therefore, fault tolerance is considered as a major concern. Because when the storage node is destroyed, all data contained in the storage node are corrupted. To solve this problem, one common solution is to introduce data redundancy. In case of the induced data redundancy approach, managing data traffic and energy consumption in storage nodes is quite challenging. Thus, in this current research network coding based routing algorithm is proposed for improving data redundancy and energy utilization of storage nodes and also improves systems reliability through finding the shortest routing path. Figure 1 sketches the architecture of the proposed data storage management system in IoT.

Four stages are involved in this proposed system. In the first stage, the deployment of a node in the harsh condition such as the earthquake, military base, hurricane, etc., is carried out and following that, the source and the destination node are fixed.

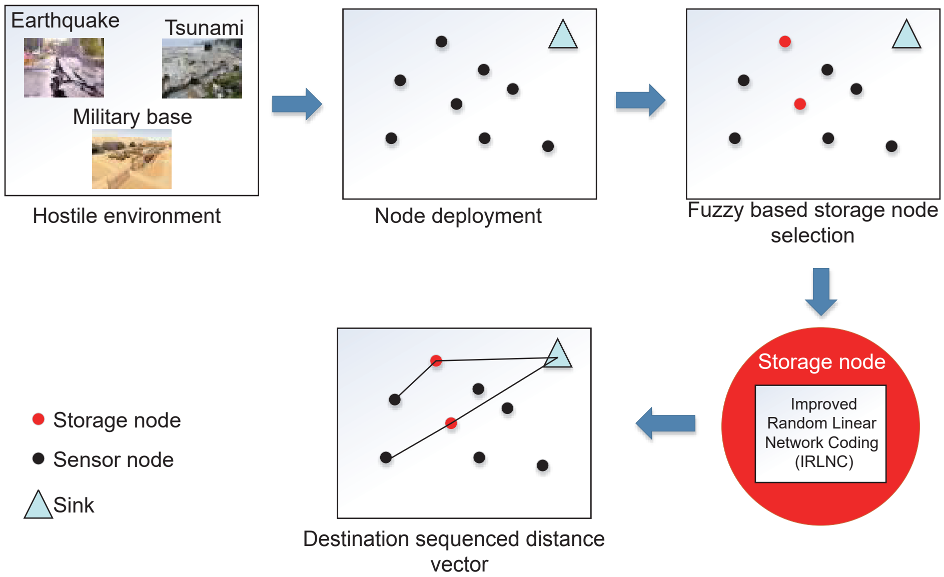

Fig. 1 Architecture of the proposed data storage management system in IoT.

Subsequently, in the second stage, the intermediate nodes present between the source and destination node that acts as storage node is selected based on considering certain essential parameters of the node such as residual energy, location of the node, workload in the node and coding capability. These essential parameters are fetched as input into the fuzzy expert system for selecting effective storage nodes to improve data storage management. Then, in the third stage, to reduce transmission delay due to the presence of huge data, the packets are encoded using coding techniques. The coding technique used in this present work for encoding the packet is IRLNC. In case of IRLNC, the concept of data compression is performed along with the process of encoding and decoding to reduce transmission overhead and improve system reliability. Finally, in the fourth stage, the transmission of data packets is achieved by means of using the shortest path to reduce energy consumption and improve network performance. Finding the shortest path is done through considering some of the parameters such as communication distance, number of received packet and packet loss. The routing algorithm used for finding the shortest path is DSDV algorithm.

3.1 Node deployment

Positioning of sensor nodes in various harsh environments is termed as node deployment. One of the major functions of these deployed nodes is to transmit the sensed information to sever for providing fast rescue operation in case of any emergency situation. In order to render efficient data transmission without any failure, an efficient algorithm that jointly optimize data storage and data routing in IoT must be designed. Designed algorithm for data storage and routing is discussed in the upcoming stages.

3.2 Storage node election

Once the nodes are deployed in the specified harsh region, the next stage is to elect efficient intermediate nodes to act as storage nodes. These storage nodes help to retrieve the data in case of any data corruption. The election of storage node is achieved through considering certain essential parameters of nodes like node location, residual energy, the workload in node and coding capability. To achieve optimal storage node election for managing data storage, fuzzy expert system is introduced. A fuzzy logic system is introduced in our proposed work because it provides greater flexibility and also has the ability to handle uncertainties in case of imprecise value. Through making alterations to fuzzy rules and membership functions, the fuzzy system can be widely utilized in various environments. In addition to its flexibility and uncertainty, it also employs human knowledge in making decisions.

3.3 Fuzzy-based storage node election

Here, the fuzzy system is employed in every node to enable them to evaluate its coding ability, residual energy and workload[20]. The evaluation is done according to rule based, which is defined on the basis of human knowledge. Normally three major steps are enclosed in fuzzy systems such as fuzzification, knowledge base and defuzzification.

3.4 Fuzzification

Fuzzification is the process of converting every input parameter into its linguistic using its corresponding membership function. The conversion of the input variable into a linguistic variable is considered as significant because only based in this fuzzification process the upcoming rule framing process is made simpler. The input parameter which is converted into its linguistic variables is residual energy, workload and coding capability. Further, the membership function used to fuzzification these input variables is the triangular membership function. A brief explanation regarding these three input variables is given below.

3.4.1 Residual energy

The remaining energy available in the node for further transmission after consumption is termed as residual energy of the node. When the node contains high residual energy, then it is said to be efficient. In case the node contains little residual energy, then the termination of the node occurs, which results in the corruption of data stored. Based on the initial energy of the node, the residual energy can be calculated using the mathematical formula given in Eq. (1).

$$\mathrm{RES} = \frac{E_{\mathrm{c}}}{E_{\mathrm{i}}} \times 10 \tag{1}$$

where RES signifies the residual energy of the node, E_{c} represents the energy consumed by the node, and E_{i} represents the initial energy of the node. This residual energy of the node is normalized within the number $[0,10]$ and it is fuzzified into three levels such as low, medium and high, and is illustrated in Fig. 2.

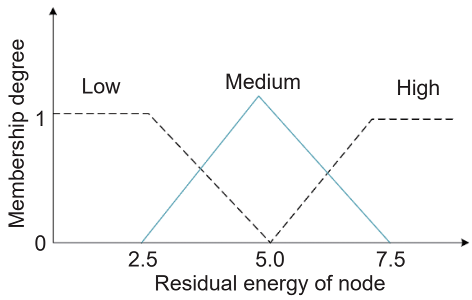

Fig. 2 Fuzzification of node residual energy.

3.4.2 Workload

The workload is calculated based on the number of packets queued in the node which is yet to be transmitted. In general, the amount of queued packets is inversely proportional to the possibility of the newly received packets being dropped in the node. Intuitively in order to reduce the chances of packet loss caused by the congested network, a node’s queue should not be overburdened by a large number of requests packets. Furthermore, a node's queue length is an appropriate measure of the available bandwidth of its outgoing connected links is indicated. To put it another way, the less congested the queue, the more bandwidth is likely to be available. The mathematical expression used for calculating workload is provided in Eq. (2).

$$\mathrm{WL} = \frac{n_{\mathrm{e}}}{c_{\mathrm{n}}} \times 100 \tag{2}$$

where WL denotes the measure of workload in the node, n_{e} signifies number of packets present in the queue, and c_{n} denotes the maximum queuing capability of the node based on the number of packets. This workload of the node is normalized within the number $[0,100]$ and it is fuzzified into three levels such as low, medium and high, and is illustrated in Fig. 3.

3.4.3 Coding capability

The coding capability of a node is calculated based on two factors as past coding status of the node and the present coding status of the node. The past coding status of the node is greatly influenced by the location of the node. If the node is present in the appropriate

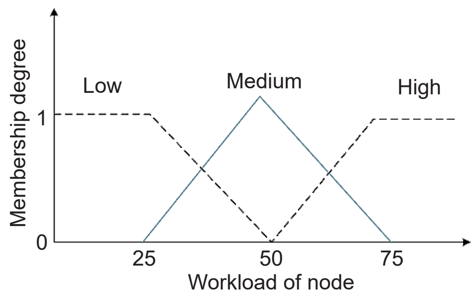

Fig. 3 Fuzzification of node workload.

region with more number of neighbors then it indicates that higher encoding is occurred in that node in the past. The past coding status of a node can be calculated using Eq. (3).

$$C_{\mathrm{h}} = \frac{n_{\mathrm{c}}}{n_{\mathrm{t}}} \times 10 \tag{3}$$

where C_{h} represents the past coding history of node, n_{c} signifies the total number of encoding occurred in the node, and n_{t} signifies the total number of packets transmitted by the node. Following that, the present coding status is influenced by means of existing packets in nodes. If the node contains a lesser number of queue packets in the buffer, then it indicates a higher possibility of encoding. The present coding status of a node can be calculated using Eq. (4).

$$C_{\mathrm{p}} = \frac{n_{\mathrm{cep}}}{C_{\mathrm{m}}} \times 10 \tag{4}$$

where C_{p} represents the present coding status of node, n_{cep} signifies number of coding eligible packet present in the queue of the node, and C_{m} signifies maximum capacity of queue based on number of packets. Finally, the overall coding capability of node is calculated as follows.

$$\mathrm{CC} = \alpha C_{\mathrm{h}} + (1-\alpha) C_{\mathrm{p}} \tag{5}$$

where CC denotes the overall coding capability of node and α is introduced to adjust the present and past coding status of the node. This coding capability of the node is normalized within the number $[0, 10]$ and it is fuzzified into three levels such as low, medium and high and it is illustrated in Fig. 4.

3.5 Knowledge base

In this knowledge base stage, the membership function that defines the input parameter is converted into a fuzzy rule. The ability to derive a solution from a set of facts is regarded as a critical feature of an expert system. In general, acquired data is transferred into a conditional statement, such as “if else”, which represents evidence. Based on the evidence knowledge gained, reasoning rules are developed, which can be either production or interference rules. Each generated rule will link two sets of factual premises and corresponding consequences. In order to describe uncertainty within knowledge, confidence factors are included alongside each fact and rule.

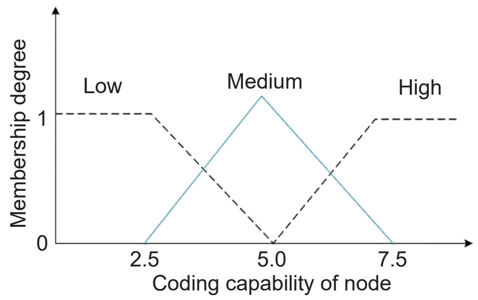

Fig. 4 Fuzzification of coding capability of node.

$$R = f(\mathrm{RES}) + f(\mathrm{WL}) + f(\mathrm{CC}) \tag{6}$$

where R denotes the storage node selection, RES denotes residual energy, WL denotes the workload, and CC determines the coding capability. Based on these function with the high election possibility can be selected as the storage node.

Defuzzification. The inverse process of fuzzification is referred to as defuzzification. It is normally the process of changing the fuzzy output to a crisp value. In other way, it can also be called as changing vague information into meaningful data. The generated output from fuzzy set is displayed between zero and one. If the obtained value is nearer to 0 then that particular node is not selected as storage node and if the obtained value is nearer to 1 then that specified node is selected as storage node.

3.6 Network coding

After the selection of an efficient storage node using fuzzy system the remaining nodes in the network transmit the received packets to the storage node.

Improved random linear network coding. Storage node stores the data packets in the form of encoded. The process of encoding is introduced in this network system in order to secure data from attackers and to improve security[21]. Coding technique used in this proposed work was IRLNC. This model is similar to the existing RLNC method but the compression of the data is an add-on contribution for the designed model, which results in improved system reliability such as packet delivery ratio and secure data with lossless transmission. A detailed description regarding IRLNC is given below. Initially in IRLNC the data is compressed at the source node prior to transmission. Let $X=(x_1,x_2,\ldots,x_n)^{\mathrm{T}}$ be the acquired data at source node during certain period of time. To compress the data a uniform random Bernoulli matrix is introduced and it is represented as B. Elements in Bernoulli matrix Bcan be represented as in Eq. (7).

$$b_{ij}=\begin{cases}-\dfrac{1}{\sqrt{n}},\ n<0.5;\\ \dfrac{1}{\sqrt{n}},\ n\geqslant 0.5\end{cases} \tag{7}$$

where n represents total collected data and $b_{ij}\in B$. Once the matrix B is determined the row in the matrix is considered a compression matrix without causing any loss to initial information. The compressed data with m dimension is represented in Z.

$$Z=BX \tag{8}$$

$$\begin{pmatrix}Z_1\\ \vdots\\ Z_m\end{pmatrix}=\begin{pmatrix}b_{11}&\cdots&b_{1n}\\ \vdots&\ldots&\vdots\\ b_{m1}&\cdots&b_{mn}\end{pmatrix}\begin{pmatrix}x_1\\ \vdots\\ x_n\end{pmatrix} \tag{9}$$

After the process of compression the source node further encodes all the data packet using RLNC technique and generates a new encoded packet[22, 23]. The process of encoding and decoding included in RLNC technique is explained below.

Encoding process. At first the input data is converted to form a matrix of size $1\times N$.

Following that the created matrix is permuted with the help of generated dynamic permutation table represented as π_{P}.

The obtained permuted data is further converted into its corresponding sub-matrices $\alpha=N/h^2$. Generated permuted sub-matrix is represented as $\mathrm{SD}=\mathrm{SD}_1,\mathrm{SD}_2,\ldots,\mathrm{SD}_\alpha$ and this permuted sub-matrix consists of h row and h column.

Next, the coefficient matrix is selected from the selection table SG $(1\times\alpha\ \text{and}\ \mathrm{NG}\leqslant\alpha)$ in a random manner. The selection table consist of random number between 1 and NG. Generated encoding matrix is represented as $G=G_1,G_2,\ldots,G_{\mathrm{NG}}$.

Further the encoded sub-matrix is generated through perform dot product between permuted sub-matrix and the coefficient matrix. Produced encoded sub-matrix is represented in Eqs. (10) and (11).

$$C_{ij}=G_{ij}\odot\mathrm{PD}_{ij} \tag{10}$$

$$C_{ij}=\begin{pmatrix}G_{11}&\cdots&G_{1h}\\ \vdots&\ldots&\vdots\\ G_{t1}&\cdots&G_{th}\end{pmatrix}\odot\begin{pmatrix}\mathrm{PD}_{11}&\cdots&\mathrm{PD}_{1h}\\ \vdots&\ldots&\vdots\\ \mathrm{PD}_{h1}&\cdots&\mathrm{PD}_{hh}\end{pmatrix} \tag{11}$$

Then, this generated coefficient matrix with size $t\times h$ must be invertible and must possess coefficient in $\mathrm{GF}(2^8)$. The final output obtained after this process is new encoded packet. The encoded packet is sent from source node to storage node.

Decoding process. Similar process is performed at the destination node to recover the original data but in the reverse order. Here, the inverse permutation table and inverse coefficient matrix is used.

Initially the sub-matrix with size $h\times h$ is extracted from the encoded matrix C_{ij} and the resulting sub-matrix is represented as Ch_{ij}.

Then, the inverse coefficient matrix is multiplied with the resulting sub-matrix. The inverse coefficient matrix is denoted as G_{ij}^{-1}.

$$PD_{ij}=G_{ij}^{-1}\odot Ch_{ij} \tag{12}$$

Finally to obtain original data the inverse permutation operation π_p^{-1} is carried out.

At this stage, the storage node contains the encoded packet received from the source node and the remaining intermediate node. Further, the encoded packet is transmitted from the storage node to the destination node using the shortest routing algorithm.

3.7 Shortest path

To transmit the encoded packet from the storage node to the destination node the shortest path is determined using DSDV algorithm. This DSDV algorithm functions on the basis of routing table[24]. The process involved in DSDV algorithm is explained as follows.

Initially, the nodes calculate its distance from its neighboring nodes and list in the routing table. Along with the distance the sequence number is also given in the table.

Sequence number represents the time at which the routing table is updated. The nodes transmit the data packet to neighboring on the basis of distance and sequence number given in the routing table.

After every data transmission, the sequence number contained in the routing table gets updated. This update ensures that every node receives correct information regarding their neighboring node.

Based on this routing protocol, the shortest path between storage node and destination node is established. Finally the encoded packets reach the destination node. At the destination node the encoded packet is decoded using the inverse process of encoding. Based on this proposed research data redundancy and system reliability can be improved which ultimately result in improved data storage management system in IoT.

4 Result and discussion

Simulation analysis on the proposed data storage management with network coding in IoT is carried out on Matlab2020b with Intel(R) Core(TM) i5-10300H processor, CPU @ 2.50 GHz, NVIDIA GTX 1650 4 GB (GDDR6) GPU, 16.0 GB memory (RAM) and system type of 64-bit operating system as system configuration. Major objective of this current research is to reduce energy consumption and improve system reliability through utilizing effective data redundancy approach. Data redundancy is maintained through utilizing fuzzy based data storage system. Then, to secure data from numerous attacks IRLNC technique is utilized and finally to reduce energy consumption shortest path is found using DSDV algorithm.

The deployment of sensor node is done within 500×500 m^2 experimental region considered for the analysis. The initial energy of the deployed node is given as 10 J. Table 1 illustrates the simulation parameter considered for the experiment.

Initially, the nodes are deployed in the specified region to establish network connectivity. Function of these deployed nodes is to gather the information for that particular surrounding and further transmit it to the sensor. Source of energy for these nodes are battery and once the energy get depleted the nodes get failed and the entire data transmission get affected. So, reducing energy consumption is considered as significance in the network system. Energy consumption in nodes can be minimized through using

Table 1 Simulation parameters considered for analysis.

Simulation parameter	Value
Number of sensor nodes	100
Simulator	Matlab R2020b
MAC type	802.11
Packet rate	100 packet/s
Simulation time	100 s
X&Y dimension	500 m & 500 m
Packet size	100 bytes
Energy	10 J
Type of channel	Wireless

shortest path for data transmission.

As the initial process in sensor network is node deployment as these nodes act as key component for data transmission. Figure 5 displays the deployment of sensor node as well as selection of source and destination node. Around 100 sensor nodes are deployed in 500 m×500 m dimensional area in a random manner. After that the selection of source and destination node is done to perform the data transmission. Green color triangle in the plot signifies source and destination node. Generally the data transmission starts from source node and terminates at destination node. Then, fuzzy based storage node selection is displayed in Fig. 6. Storage node is selected through incorporating fuzzy concept and providing residual energy, workload and coding capability as input. The storage node stores the data gathered from the neighboring node to overcome data corruption issue. Red color circle in the plot describes the selected storage node.

As the source node and all other neighboring node transmit the data to the storage node following that the storage node must transmit the data packet finally to

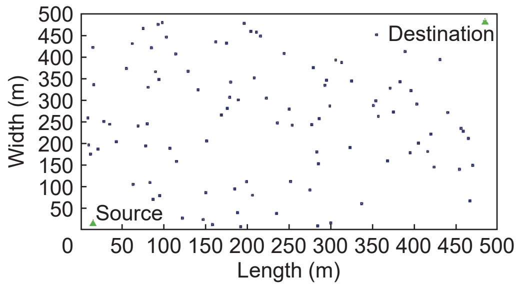

Fig. 5 Node deployment and selection of source and destination node.

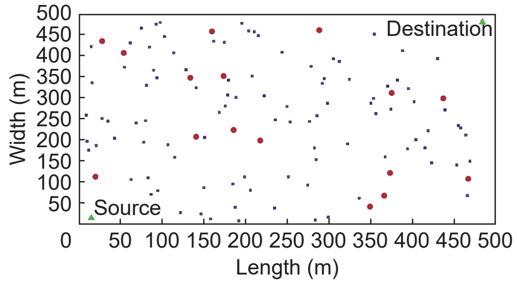

Fig. 6 Fuzzy based storage node election.

destination node. Figure 7 illustrates the selection of shortest path between source and destination. Finding shortest path is achieved using DSDV algorithm. Purple color nodes in the plot displays the path selected for data transmission. Moreover the performance of the current research is estimated through calculating some of the performance metrics such as delay, Packet Delivery Ratio (PDR), Throughput, energy consumption, encoding time, decoding time, execution time and storage time. Along with that the performance of the proposed (Fuzzy + IRLNC) is compared with some of the existing techniques like Active Intersession Coding Aware Routing (AINC-AR)[24], Distributed Coding Aware Routing (DCAR)[25], Backpressure based NC-aware Routing (BPNCR)[26], and Connected dominating set-based and Flow-oriented Coding-aware Routing (CFCR)[23].

Delay comparison between proposed and existing techniques based on number of nodes is displayed in Fig. 8. For 100 nodes the value of delay obtained for the proposed Fuzzy+ IRLNC is 75 s and for existing techniques such as AINC-AR, DCAR, BPNCR and CFCR the delay value is determine to be 100 s, 150 s, and 200 s. whereas in case of 500 nodes the delay for proposed and existing techniques such as AINC-AR,

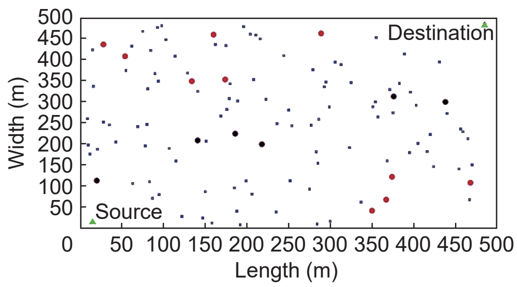

Fig. 7 Selection of shortest path between source and destination.

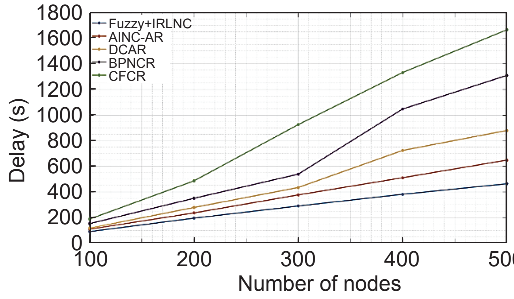

Fig. 8 Delay comparison based on number of nodes.

DCAR, BPNCR, CFCR was determine to be 425 s, 620 s, 850 s, 1300 s, and 1650 s. This delay comparison analysis proves that lesser delay value is attained by the proposed work on comparison with existing.

Value obtained for the throughput metric is compared between the proposed and existing technique and displayed in Fig. 9. The graph in this figure is plotted between number of nodes from 100–500 and throughput value in Megabits Per Second (Mbps) on both axes. Proposed Fuzzy+IRLNC method achieves 0.132 Mbps in case of 100 nodes, whereas the existing techniques such as AINC-AR, DCAR, BPNCR, and CFCR attain 0.121 Mbps, 0.117 Mbps, 0.114 Mbps, and 0.11 Mbps. Similarly in case of 500 nodes the throughput for proposed and existing techniques such as AINC-AR, DCAR, BPNCR, and CFCR was determine to be 0.16 Mbps, 0.144 Mbps, 0.139 Mbps, 0.138 Mbps, and 0.136 Mbps. The greater value for throughput proves that the proposed work is effective on comparison with existing in IoT platform.

Average residual energy reached for the proposed method is compared with some of the existing techniques such as AINC-AR, DCAR, BPNCR, and CFCR, as given in Fig. 10. For this figure the graph is drawn between various techniques on the *X*-axes and average residual energy in percentage on the *Y*-axes.

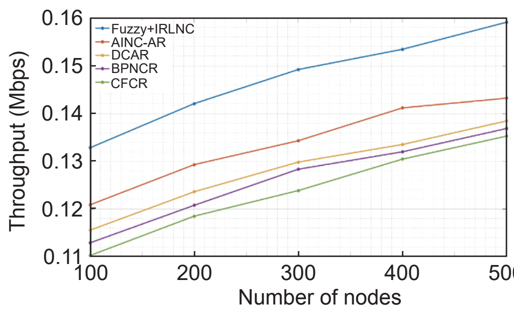

Fig. 9 Throughput comparison based on number of nodes.

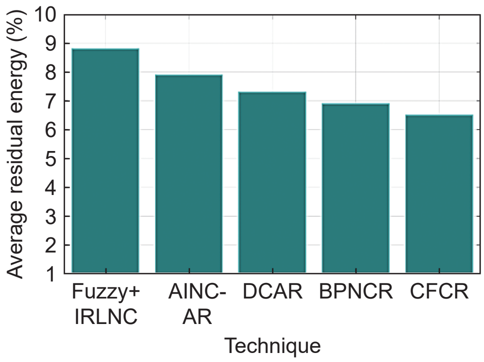

Fig. 10 Comparison of average residual energy.

8.8% is the average residual energy attained for the proposed Fuzzy+ IRLNC technique On the other hand 7.9%, 7.4%, 6.9%, and 6.6% is the average residual energy achieved by the existing technique such as AINC-AR, DCAR, BPNCR, and CFCR. This higher rate of residual energy for the proposed work proves its effective functioning when compared to existing.

Comparison of performance between proposed and existing techniques in terms of PDR is given in Fig. 11. PDR is the ratio between numbers of packet transmitted successfully to the total number of packets. Value of PDR obtained for proposed Fuzzy+ IRLNC is 0.92%. On the other hand the PDR value reached for existing AINC-AR, DCAR, BPNCR, and CFCR is 0.84%, 0.81%, 0.74 %, and 0.69%. This higher rate of PDR for the proposed work proves its effective functioning when compared to existing.

Packet loss comparison between proposed and existing techniques is given in Fig. 12. Packet loss is the ratio between numbers of packet lost during transmission to the total number of packets. Value of packet loss obtained for proposed Fuzzy+ IRLNC is 0.075%. On the other hand the packet loss value

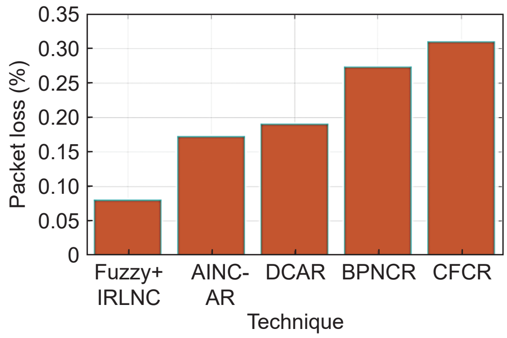

Fig. 12 Comparison of packet loss.

obtained for existing AINC-AR, DCAR, BPNCR, and CFCR is 0.17%, 0.19%, 0.27%, and 0.32%. This lesser rate of packet loss for the proposed work proves its effective functioning when compared to existing.

Comparing the performance between proposed and existing techniques in terms of compression ratio is given in Fig. 13. Compression ratio is nothing but the length of uncompressed data divided with length of compressed data. Value of compression ratio obtained for proposed Fuzzy+ IRLNC is 0.82 %. On the other hand the compression ratio reached for existing AINC-AR, DCAR, BPNCR, and CFCR is 0.79 %, 0.71%, 0.65%, and 0.60%. This higher rate of compression ratio for the proposed work proves its effective functioning when compared to existing.

Analyzing the value of information loss achieved by the proposed Fuzzy+ IRLNC method and further comparing it some of the existing techniques is displayed in Fig. 14. Information loss is the ratio of data retained in the encoded packet to the data contained in the original packet. The proposed Fuzzy+ IRLNC achieves 0.02% information, whereas the existing techniques such as AINC-AR, DCAR,

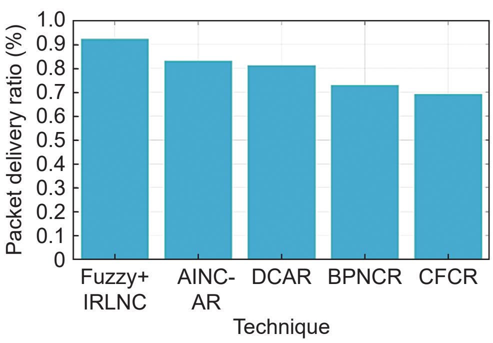

Fig. 11 Comparison of packet delivery ratio.

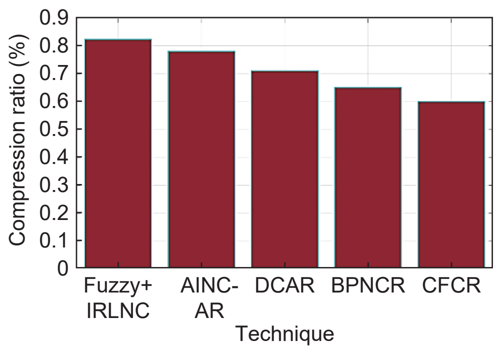

Fig. 13 Comparison of compression ratio.

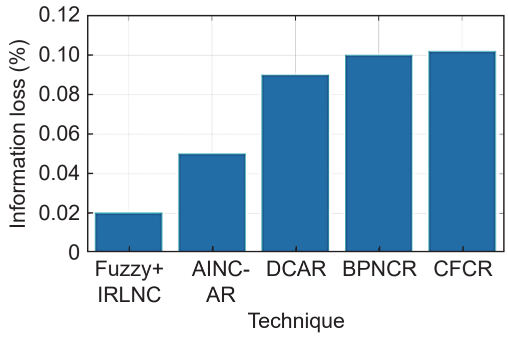

Fig. 14 Comparison of information loss.

BPNCR, and CFCR attains 0.05%, 0.09%, 0.1%, and 0.12% as information loss. This comparison analysis shows that low information loss is reached by the proposed method when compared to existing.

Figure 15 illustrates the comparison analysis performed in-between proposed and existing based on encoding time. Encoding time is the time taken for encoding the data packets. Value of encoding time obtained for proposed Fuzzy+ IRLNC is 350 s. On the other hand the encoding time reached for existing AINC-AR, DCAR, BPNCR, and CFCR is 420 s, 475 s, 500 s, and 5 s. This comparison analysis shows that lesser encoding time is reached by the proposed method when compared to existing.

Performance attained by the proposed method is compared with some of the existing technique based on the statistical parameter which is decoding time. This decoding time comparison study is illustrated in Fig. 16. Decoding time is the time taken for decoding the encoded packet to reach the original packet. 375 s is the value of decoding time reached for proposed Fuzzy+ IRLNC. Following that 420 s, 460 s, 475 s, and 520 s is the decoding rime reached for existing AINC-AR, DCAR, BPNCR, and CFCR techniques. Based on

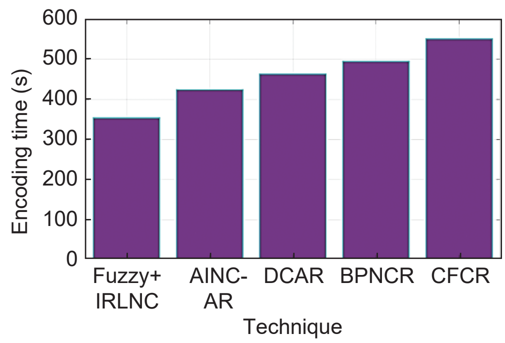

Fig. 15 Comparison of encoding time.

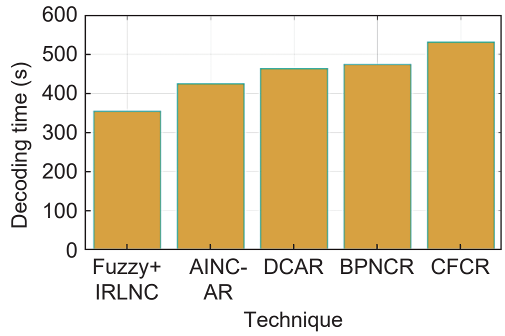

Fig. 16 Comparison of decoding time.

this study it is found lesser decoding time is reached for the proposed method when compared to existing.

Comparison study carried out based on execution time for the proposed and existing routing techniques is displayed in Fig. 17. Execution time is the total time taken for the entire simulation analysis. When compared to other existing techniques such as AINC-AR, DCAR, BPNCR, and CFCR the proposed method attains lesser execution time of about 700 s. the execution time for the proposed method is determine to be 820 s, 960 s, 980 s, and 1000 s for AINC-AR, DCAR, BPNCR, and CFCR respectively. Effective performance is achieved by the proposed method when compared to existing technique as it attains lesser execution time.

Storage time is nothing but the time taken for storing the encoded data packets in storage node. This value of storage time obtained for the proposed Fuzzy+ IRLNC is compared with existing techniques and given in Fig. 18. From the graph, it is found that 69 s is the storage time gained for proposed Fuzzy+ IRLNC method. Following that 78 s, 89 s, 92 s, and 98 s is the storage time attained by the existing AINC-AR, DCAR,

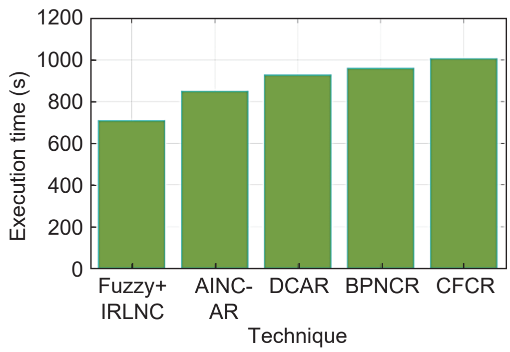

Fig. 17 Comparison of execution time.

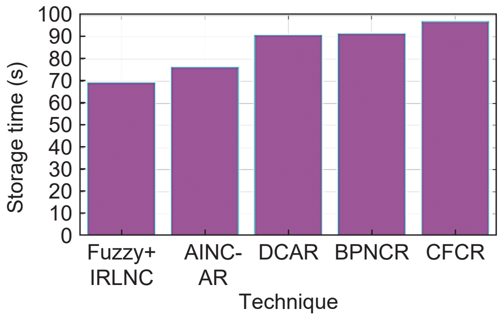

Fig. 18 Comparison of storage time.

BPNCR, and CFCR techniques.

In Table 2, some of the effective routing algorithms are compared with the proposed model such as Trustworthy Cluster-based Energy and Lifetime conscious Routing (TCELR)[27], Density-based Fuzzy C-Means clustering (DFCM)[28], and Cluster-based Backpressure Routing (CBPR)[29]. From this evaluation, the performance of the proposed method is proved to be effective when compared to existing techniques as it achieves high residua energy, low throughput and packet delivery ratio.

5 Conclusion

This current research focuses on jointly optimizing both the storage problem and system reliability in IoT platforms. Generally, IoT based sensor network is utilized for automatic monitoring of the specified area and helps in converting them to a modernized one. However, data corruption and the unreliable system is a major issue due to frequent node failure because of deployment in harsh environment. To minimize data corruption designated storage system must be introduced in the network system. On the other hand, to improve system reliability, efficient network coding and optimal route selection must be included. So, in the proposed work, initially, the sensor nodes are positioned in harsh environment. Then, to store the data, effective intermediate nodes are selected as storage nodes based on fuzzy system. Residual energy, workload and coding capability are provided as input to generate fuzzy rules. Following that, the data packet is encoded using IRLNC technique. In case of IRLNC, the data are compressed prior to encoding. These encoded packets stored in the storage node are transmitted to the destination node based on the DSDV algorithm. Experimental analysis of the proposed system in IoT platform is performed using statistical metrics and further compared with existing techniques to show the better functioning of the proposed research and its suitability in real time IoT application.

Table 2 Comparison of proposed and existing routing algorithm.

Algorithm	Residual energy (J)	Throughput (Mbps)	PDR (%)
Fuzzy+IRLNC	9.5	0.13	0.92
TCELR	10	0.05	0.74
DFCM	9	0.08	0.86
CBPR	12	0.035	0.88

References

[1] K. Haseeb, I. Ud Din, A. Almogren, and N. Islam, An energy efficient and secure IoT-based WSN framework: An application to smart agriculture, *Sensors*, vol. 20, no. 7, p. 2081, 2020.

[2] K. Haseeb, A. Almogren, N. Islam, I. Ud Din, and Z. Jan, An energy-efficient and secure routing protocol for intrusion avoidance in IoT-based WSN, *Energies*, vol. 12, no. 21, p. 4174, 2019.

[3] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, I-SEP: An improved routing protocol for heterogeneous WSN for IoT-based environmental monitoring, *IEEE Internet Things J.*, vol. 7, no. 1, pp. 710–717, 2020.

[4] A. H. Bagdadee, M. Z. Hoque, and L. Zhang, IoT based wireless sensor network for power quality control in smart grid, *Procedia Comput. Sci.*, vol. 167, pp. 1148–1160, 2020.

[5] K. Haseeb, N. Islam, A. Almogren, I. Ud Din, H. N. Almajed, and N. Guizani, Secret sharing-based energy-aware and multi-hop routing protocol for IoT based WSNs, *IEEE Access*, vol. 7, pp. 79980–79988, 2019.

[6] S. Kumar and V. K. Chaurasiya, A strategy for elimination of data redundancy in Internet of Things (IoT) based wireless sensor network (WSN), *IEEE Syst. J.*, vol. 13, no. 2, pp. 1650–1657, 2019.

[7] O. I. Khalaf and G. M. Abdulsahib, Optimized dynamic storage of data (ODSD) in IoT based on blockchain for wireless sensor networks, *Peer Peer Netw. Appl.*, vol. 14, no. 5, pp. 2858–2873, 2021.

[8] S. P. Sasirekha, A. Priya, T. Anita, and P. Sherubha, Data processing and management in IoT and wireless sensor network, *J. Phys.: Conf. Ser.*, vol. 1712, no. 1, p. 012002, 2020.

[9] O. Diallo, J. J. P. C. Rodrigues, and M. Sene, Real-time data management on wireless sensor networks: A survey, *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 1013–1021, 2012.

[10] B. D. Deebak and F. Al-Turjman, A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks, *Ad Hoc Netw.*, vol. 97, p. 102022, 2020.

[11] S. Sankar, P. Srinivasan, S. Ramasubbareddy, and B. Balamurugan, Energy-aware multipath routing protocol for Internet of Things using network coding techniques, *Int. J. Grid Util. Comput.*, vol. 11, no. 6, pp. 838–846, 2020.

[12] C. H. S. Oliveira, Y. Ghamri-Doudane, C. E. F. Brito, and S. Lohier, Optimal network coding-based In-network data storage and data retrieval for IoT/WSNs, in *Proc. IEEE 14th Int. Symp. Network Computing and Applications*, Cambridge, MA, USA, 2015, pp. 208–215.

[13] S. Malathy, V. Porkodi, A. Sampathkumar, M. H. D. N. Hindia, K. Dimyati, V. Tilwari, F. Qamar, and I. S. Amiri, An optimal network coding based backpressure routing approach for massive IoT network, *Wirel. Netw.*, vol. 26, no. 5, pp. 3657–3674, 2020.

[14] C. Chen, L. Zhang, and R. L. K. Tiong, A new lossy compression algorithm for wireless sensor networks using Bayesian predictive coding, *Wirel. Netw.*, vol. 26, no. 8, pp. 5981–5995, 2020.

[15] Z. Li, M. Xu, T. Liu, and L. Yu, A network coding-based braided multipath routing protocol for wireless sensor networks, *Wirel. Commun. Mob. Comput.*, vol. 2019, p. 2757601, 2019.

[16] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, On randomized network coding, in *Proc. 41th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, USA, 2003, pp. 11–20.

[17] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, The benefits of coding over routing in a randomized setting, in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, Japan, 2003, p. 442.

[18] N. Chervyakov, M. Babenko, A. Tchernykh, N. Kucherov, V. Miranda-López, and J. M. Cortés-Mendoza, AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, *Future Gener. Comput. Syst.*, vol. 92, pp. 1080–1092, 2019.

[19] H. Yang, F. Li, D. Yu, Y. Zou, and J. Yu, Reliable data storage in heterogeneous wireless sensor networks by jointly optimizing routing and storage node deployment, *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 230–238, 2021.

[20] P. Rafiee and G. Mirjalily, Distributed network coding-aware routing protocol incorporating fuzzy-logic-based forwarders in wireless ad hoc networks, *J. Netw. Syst. Manag.*, vol. 28, no. 4, pp. 1279–1315, 2020.

[21] H. N. Noura, R. Melki, M. Malli, and A. Chehab, Design and realization of efficient & secure multi-homed systems based on random linear network coding, *Comput. Netw.*, vol. 163, p. 106886, 2019.

[22] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, The role of caching in future communication systems and networks, *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1111–1125, 2018.

[23] H. Alshaheen and H. Takruri-Rizk, Energy saving and reliability for wireless body sensor networks (WBSN), *IEEE Access*, vol. 6, pp. 16678–16695, 2018.

[24] D. Sinwar, N. Sharma, S. K. Maakar, and S. Kumar, Analysis and comparison of ant colony optimization algorithm with DSDV, AODV, and AOMDV based on shortest path in MANET, *J. Inf. Optim. Sci.*, vol. 41, no. 2, pp. 621–632, 2020.

[25] J. Chen, K. He, R. Du, M. Zheng, Y. Xiang, and Q. Yuan, Dominating set and network coding-based routing in wireless mesh networks, *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 423–433, 2015.

[26] Z. Mei and Z. Yang, Active intersession network coding-aware routing, *Wirel. Netw.*, vol. 23, no. 4, pp. 1161–1168, 2017.

[27] G. Vasanthi and N. Prabakaran, TCELR: Trusted cluster based energy and lifetime aware routing protocol for wireless sensor network using hybrid bird swarm-differential search algorithm, *Int. J. Adv. Sci. Technol.*, vol. 29, no. 12s, pp. 892–913, 2020.

[28] D. Kalaimani, Z. Zah, and S. Vashist, Energy-efficient density-based Fuzzy C-means clustering in WSN for smart grids, *Aust. J. Multi-Discipl. Eng.*, vol. 17, no. 1, pp. 23–38, 2021.

[29] R. Maheswar, P. Jayarajan, A. Sampathkumar, G. R. Kanagachidambaresan, M. H. D. Nour Hindia, V. Tilwari, K. Dimyati, H. Ojukwu, and I. S. Amiri, CBPR: A cluster-based backpressure routing for the Internet of Things, *Wirel. Pers. Commun.*, vol. 118, pp. 3167–3185, 2021.