Journal of Engineering Science and Technology
Vol. 16, No. 6 (2021) 5087 - 5109

PERFORMANCE OPTIMISATION WITHIN DEVICE LAYER OF IOT NETWORKS

K. V. SOWMYA[1], J. K. R. SASTRY[2,*]

[1]Department of Electronics and Communication Engineering, KLEF University, Vaddeswaram, Guntur District, Andhra Pradesh, India
[2]Department of Electronics and Computer Engineering, KLEF University, Vaddeswaram, Guntur District, Andhra Pradesh, India
*Corresponding Author: drsastry@kluniversity.in

Abstract

The performance of IoT networks greatly changes when different networking topologies are used within different layers of the IoT network. The performance of the Device layer is dependent on the way the devices are connected, the cluster head is elected through the use of an appropriate clustering algorithm, the extent of the existence of heterogeneity among communication protocols, the level of fault tolerance, longevity of the devices managed through power management, etc. A model is presented that implements a Modified Low energy adaptive clustering hierarchy(LEACH) Algorithm that combines cluster head selection mechanism, Power Minimisation, handling heterogeneity, and ensuring a high degree of fault tolerance. The model is implemented on a prototype model and found to perform 15% more than the other methods used for implementing Clustering within the device level.

Keywords: Clustering device, IoT networks, Networking devices, Performance optimisation.

1. Introduction

The Internet of things is a new paradigm where it has led to a proliferation of new devices. The information is gathered from the physical world through the sensors, and these devices, in turn, connect to the network to provide the information they gathered. An IoT network can be divided into three main components i) Device, ii) Gateway, iii) Cloud, each of which plays an important role in making a successful network.

Wireless sensor networks (WSN) are an integral part of IoT, where a group of sensors will be deployed in an area to measure the parameters of the physical surroundings. The data collected by the sensors will be sent to a cloud through a gateway [1]. A Gateway acts as a bridge between a services server and a cloud where the data is stored for analysis and decision making.

A typical IoT network will have several layers of networking. The performance of the network at each of the layers shall have an overall impact on the entire network. Many topologies can be used for building an IoT based network. In each of the Layer, different networking mechanisms can be used for achieving the interconnection between the devices. The commonly used networking mechanisms that can be used for establishing networking include Point to Point networking, networking as a Star, Mesh, Bus, Butterfly, Crossbar, Multi-Stage, or a bus-based network [2].

Clustering is a way of connecting together a set of devices using a networking topology generally using a wireless Communication technology and communication from each of the device is undertaken through a chosen device called cluster head

In a cluster, the connectivity of the devices is a critical issue. Appropriate and efficient network / Topology are to be implemented within this Layer considering the performance of the same. At the same time, a high level of fault Tolerance of the same is achieved.

Communication among the devices takes place, considering a set of protocols. The use of only one protocol makes things easier, and the performance of communicating is much higher. In contrast, the performance of the clusters suffers a lot when heterogeneous protocols are used within the devices for effecting the communication. In such a case, the use of gateways or the protocol converters would become necessary, which generally leads to performance degradation [3].

The longevity of the devices is dependent on the power utilized for sensing and transmission of the sensed data. The total power required for data capturing and transmission should be as minimum as possible so that more amount of the data can be transmitted using basic power sources without the need for energizing the same for further functioning of the devices.

The devices in the cluster can individually talk to a base station, in which case too many channels will be required for communication, and also, the base station will be congested with a heavy load. On the other hand, one device in the cluster acting as head for communication makes the communication process much easier [4]. While that being the case, there is a necessity to have an efficient algorithm that selects a device as a cluster head, which is used for effecting communication with the base station for moving the data from a device uphill to the cloud computing system.

Sometimes a single cluster head is also risky as the failure of the same will lead to the failure of the entire cluster. Sometimes the use of a cluster of cluster heads leads to highly resilient IoT networks. Thus a model is required that considers different networking considering the issue of fault tolerance, Effective communication among heterogeneous devices, selection of a cluster head, and the use of minimum power for transmission.

The performance of an IoT network depends on several factors, which include protocol used, transmission speeds, the latency of the device, availability of alternate paths for communication, the existence of heterogeneity, type of devices used for communication, the way IoT networking is done, Conversion of service to commands and vice versa, etc. The performance of an IoT network could be considered as a summation of performance in each of the network layers and the interconnection between the layers. The total performances of the network can be computed as follows:

Performance of IoT network = performance at device level + Performance at Controller level _+ performance at services level + performance at gateway level + performance at Cloud level

Therefore, it is a clear case that performance at an IoT layers needs to be analysed independently and also in conjunction with its superseding Layer.

Performance of an IoT network can be measured with the help of metrics that include response time, throughput, data transmission rates, which all depend upon the availability of Bandwidth, Latency of devices, the existence of Jitter, Error Rate, Packet loss, heterogeneity in protocols used for effecting communication, frequent conversions, use of too many gateways, availability of low power with the devices, etc.

For this research, response time is used as a measure of the performance of an IoT network. The response time takes into account the time taken to receive the packets, buffering, de-pocketing, and pocketing [28]. Response time is computed at every Layer of the IoT network and added to get the overall response time of the entire IoT network.

As the number of IoT devices being connected to the Internet is increasing every day, the load on different networking and processing devices is getting increased tremendously, resulting in performance degradation of the whole network. One solution to this problem is to carry the Clustering of the devices and electing a cluster head that collects all the data from the sensors and transmits them to the server. The data from the sensors is processed at the cluster head and transmitted to the server. The clustering technique increases the network lifetime, energy efficiency, throughput, scalability, bandwidth usage and decreases the network congestion,

The load on the server heavily increases when each of the devices collecting the data communicates independently, notwithstanding the other. Clustering the device and electing a cluster head that communicates with the controller will enable single-channel transmission leading to a reduction of overhead on the server. At the same time, the system can become fragile and lead failures.

Clustering involves three aspects that include networking, selection of a Cluster Head, and protocol conversion when communication between heterogeneous devices is needed [5]. Clustering and electing a Cluster head also depletes the lifetime of the Cluster Head and also involves single-channel transmission. Failure of the single channel leads to complete disruption in the communication of the sensed data. A group of cluster heads sometimes enhances the performance of the IoT networks, which is also one of the main focuses of this issue.

Many protocols are being used for effecting the communication between the devices that are used for collecting and transmitting the data to a higher level of the network. The performance of transmission of the collected data is very much dependent on the type of protocols; while some being lightweight, some are heavily guarded. If the protocol used for transmission is different from the receiving protocol, then a translation of data has to be done and transmitted to the receiving protocol either through a software translator or through a Hardware-based Gateway or the translation undertaken through software implemented within a gateway. The process of the translation done either using a gateway or through the software or the software implemented in the gateway consumes some time, which reflects on the response time of the IoT network.

The IoT devices are remotely situated and generally powered through batteries. The longevity of the battery, thereby, the device is dependent on the amount of power used for transmission of the collected data. The less the power used, the more would be the life of the device. When less power is used, transmission rates will fall, leading to the issue of degraded performance. The power attached to the devices must be managed in such a way that the transmission is undertaken for a long period and with moderate speeds. Total data transmitted through a device for a given power source must be maximum.

Fault tolerance of a network built within a cluster is one of the most crucial issues; the performance is dependent on the availability of a path for transmission of the data even when some failures of the devices and network links happen [6]. If no path is available for transmission of the data, the performance of the network is bound to fall. The availability of slow-speed links is also a matter of concern.

A node within a cluster with more number of resources is generally selected as a cluster head, which in a way can be considered as a kind of gateway of the cluster. All the information regarding that particular cluster is forwarded through the gateway to the higher level communicating device. Any problem in the gateway functioning will affect outgoing communication in terms of speed of data communication. The performance of communication at the gateway may also suffer due to several reasons, which include protocol conversion, inappropriate communication standards used, mismatch in the communication interface, and the kind of media used for transmission of data

Some protocols or architectures used in the IoT system may have scalability problems i. e they cannot adapt to the network if there is an increase in the number of nodes due to which the performance of the IoT network suffers. Bandwidth consumption is one more bottleneck i. e if any of the cluster head consumes more bandwidth than the expected to transmit the information from the source to sink, the performance of the IoT network will affect.

Transmission time i. e the amount of time taken for the data packets to travel from the source to the destination also acts as one of the bottlenecks. If the time consumed is more, ultimately, the performance of the IoT network is degraded.

A number of issues are to be considered that effect performance of an IoT network within the Device layer of an IoT network. In this paper, the issues related to power dissipation, communication, fault tolerance, clustering and cluster head selection method have been discussed that together enhance the performance of an IoT network.

2. Related Work

Performance of an IoT Network depends primarily in the networking layer into which the devices are placed, networked and made to communicate. Clustering of the devices is the key.

A method is proposed by Khediri et al. [7] called “Optimized Low Energy Adaptive Clustering Hierarchy (OLEACH),” which reduces the energy consumed by the devices to a greater extent. Mamta [8] have described various clustering techniques that have been presented in the literature. A comparative picture of the same is provided.

Singh and Sharma [9] have proposed an Adaptive Data-centric Clustering algorithm for Sensor networks (ADCS), a hierarchical algorithm where user-specific data requirements are factored into the clustering decisions. Specifically, the similarity in parameter variations is used as a criterion for optimization.

Wang et al. [10] have presented a network that aims for the longevity of a IoT network. This protocol proposes an improved rotation mechanism of the cluster head based on LEACH through the adaption of tokens that minimize the reconstruction of clusters and change the cluster head in the original cluster.

Mahajan and Dhiman [11] presented an introduction to Clustering and various issues that include CH Rotation and replacement, Inter-cluster, and intra-cluster communication for effecting device-based Clustering. Kwon and Park [12] presented the way Optimum cluster size that can yield the highest transfer rate given an encoding scheme can be determined.

Xu et al. [13] have conducted a review/survey on existing protocols, which are analysed from a QoS perspective, which includes three main objectives of energy efficiency, reliable communication, and latency awareness.

A method is proposed by Puschmann et al. [14], which can be used to determine the number of clusters based on the collected stream data and the way the data is distributed. The method proposed by them adapts itself as the data changes, which is one of the characteristics of an IoT network.

Ju and Zhang [15] proposed an adaptive clustering framework for battery-less things (IoBT) nodes within an IoT network. The framework combines the optimal cluster selection method and a lexicographic optimal rate system that enables each node to utilize the scavenged energy and achieve a high transmission rate fully.

An algorithm called SSNC (smart sensor networking using clustering approach) is proposed by Ananthi et al. [16] that aim to reduce energy consumption through the replacement of dead clusters that exists between the cluster members. Fouladlou and Khademzadeh [17] proposed a new energy-efficient clustering algorithm for grouping sensor devices in IoT. The protocol is implemented through use of genetic algorithm for Clustering to decrease the energy consumption of power-limited wireless nods.

Bhandari et al. [18] proposed a clustering algorithm that takes the devices related Information (residual energy, closeness factor) available in the gateway, which is downloaded from the cloud. The proposed method significantly prolongs the network lifetime and enhances the overall throughput of the system.

Choi et al. [19] proposed the Cluster CoAP scheme for message Queuing as an extension to CoAP. They have implemented a testbed of Cluster Constrained Application Protocol Message Queue (CoAP)MQ, and the performance analysis was made with a testbed implementation. From the experimental results, we can see that the proposed scheme provides better performance than the existing MQTT and CoAP schemes.

An algorithm called LEACH (Least energy adapted cluster hierarchy LEACH) is proposed by Ashwini and Rakesh [20], which aims to elect a different cluster head for every round so that the energy balancing is done in a right way.

Al-Janabi and Al-Raweshidy [21] proposed SDN based architecture using which the load on the IoT network can be adjusted. A centralized SDN controller uses the resources, i.e., data centres and storage units existing in the cloud. An algorithm PSO (Particle Swarm Optimization) is implemented within the controller for calculating the load-balancing, communication cost, and remaining energy factors to construct a clustering table (CT).

Sung et al. [22] proposed a multi-hop clustering mechanism for IoT networks to minimize the number of required Internet connections. The main objective of their algorithm is to select a minimum number of coordinators, which take the role of a representative node for the cluster, i.e., having the Internet connection on behalf of the rest of the nodes in the cluster and to map a partition of the IoT nodes onto the selected set of coordinators to minimize the total distance between the nodes and their respective coordinator under a certain constraint in terms of maximum hop count between the IoT nodes and their respective coordinator. They have pursued a heuristic approach to solve the problem and analyse the complexity of the proposed solution.

Nguyen et al. [23] proposed software-defined Networking (SDN) of IoT networks (SDIoT). In particular, they have presented the design of a streamline SD-IoT controller, a lightweight and reconfigurable software-defined virtual sensor (SDV Sensor) to represent its underlying sensors

Wang et al. [24] presented a novel approach called as machine type communication (MTC) in which the devices can sense the surrounding environment and communicate the same data to the base stations (BS) for further analysis. As the devices that are deployed for any application are more in number, there is a need to make those devices as a cluster and pre-process the redundant data to avoid traffic overload. Another major hindrance in IoT networks is the limited battery capacity of the devices, which gives rise to an energy cost problem. An uneven cluster formation is proposed for load balancing and energy efficiency. To balance the energy consumption in each cluster, a distributed cluster head rotation mechanism is proposed.

Kumar and Zaveri [25] introduced an application-based two-layer architectural framework for IoT, which consists of a sensing layer and an IoT layer.

Lin et al. [26] have presented a device-level multimodal data correlation mining model designed based on the canonical analysis to transform the data feature into a subspace and analyse the data correlation.

Behera et al. [27] presented modification to the existing low-energy adaptive clustering hierarchy (LEACH) clustering through introducing a threshold limit for cluster head selection with simultaneously switching the power level between the nodes so that a node is selected as a cluster head for a certain limited number of times so that the chances of the same node being selected as a cluster head can be avoided.

Choi et al. [28] proposed an enhanced cluster based CoAP. The scheme proposed by them is different from the existing cluster based CoAP scheme in terms of providing cache memory in the cluster head for the efficient gathering of sensing data.

Behera et al. [29] focused on an efficient cluster head election scheme that rotates the cluster head position among the nodes with higher energy levels as compared to others. The algorithm considers initial energy, residual energy, and an optimum value of cluster heads to elect the next group of cluster heads for the network that suits for IoT applications such as environmental monitoring, smart cities, and systems.

Many authors have dealt with the issue of improving the performance of the IoT Networks either by stating key issues, performance bottlenecks, Connectivity, Fault Tolerance, the extent of communication channels and protocols [30-38].

3. Comparative Analysis and Fixation of the GAP

A comparison of various characteristics used by different authors is shown in Table 1. It is seen from this table that none have considered the issue of heterogeneity and the way to establish different communication paths so that parallel transmission of the data from the sensors can be transmitted. Most of the algorithms have considered a single cluster for transmission of the data from a set of devices. Failure of a cluster device generally leads to failure of the entire cluster as a whole.

4. Prototype IoT Model

A typical IoT network developed for carrying the experimentation is shown in Fig. 1. The IoT network has been built considering all the layers situated in a typical and comprehensive IoT network that includes the device layer, controller layer, services layer, gateway layer, and computing layer, and the Devices in the device layer are connected as a cluster.

Four clusters are included in the device layer. The first clusters contain three temperature sensors that are connected completely with an elected Cluster head, which communicates with a base station. There are three more clusters similar to the Temperature sensors, which include a Humidity Sensing cluster, Air-condition Cluster, and a FAN cluster, each communication through its cluster head with the Base Station. Each cluster is made of 4 devices, which all communicate using wifi with the base station. In the network, every device is connected to every other device. There are three paths for every device to get connected with the Cluster Head.

In the next Layer, the base station is connected to a Controller in a peer to Peer to connection, and the controller is connected to a restful services server using a peer again to peer connection. The services server keeps the status of the device. It

Table 1. Characteristics based comparison of clustering algorithms.

No.	Method	Bandwidth	load to a sensor	Throughput	Network Lifetime	Response Time	Power Consumption	Packet drop probability	Energy Efficiency	End to End delay	Packet delivery ratio	Size of the cluster	Clustering Networks	Heterogeneity
1	Enhanced Cluster-based CoAP	√	√	X	X	X	X	X	X	X	X	X	**X**	**X**
2	I-LEACH	X	X	√	√	X	X	X	X	X	X	X	**X**	**X**
3	Distributed cache strategy for an analytic cluster	X	X	X	X	√	X	X	X	X	X	X	**X**	**X**
4	Clustering algorithms based on heuristic and graph	X	X	X	√	X	√	X	X	X	X	X	**X**	**X**
5	Centralized detection system	X	X	X	X	X	X	√	X	X	X	X	**X**	**X**
6	An energy-efficient clustering routing algorithm	X	X	√	√	X	X	X	√	X	X	X	**X**	**X**
7	Software-defined IoT(SDIoT)	X	X	X	X	√	X	X	X	X	X	X	**X**	**X**
8	C-LBCA algorithm	X	X	√	√	X	X	X	√	X	X	X	**X**	**X**
9	LEACH	X	X	X	X	√	√	X	X	X	X	X	**X**	**X**
10	Cluster-based COAP	√	X	X	X	√	X	X	X	X	X	X	**X**	**X**
11	Cloud-Assisted Device Clustering	X	X	√	√	X	X	X	X	X	X	X	**X**	**X**
12	Energy Efficient Clustering Algorithm	X	X	√	X	X	X	X	√	√	X	X	**X**	**X**
13	SSNC (smart sensor network Using Clustering approach)	X	X	X	X	X	X	X	√	√	√	X	**X**	**X**
14	Adaptive Clustering	X	X	√	X	X	X	X	X	X	X	X	**X**	**X**
15	Approximate decoding approach	X	X	X	X	X	X	X	X	X	X	√	**X**	**X**
16	Packet Discarding based on Node Clustering (PDNC)	X	X	X	X	X	√	X	X	√	√	X	**X**	**X**
17	Improved rotation mechanism of cluster head based on LEACH	X	X	X	X	X	√	X	X	X	X	X	**X**	**X**
18	O-LEACH	X	X	X	√	X	X	X	X	X	X	X	**X**	**X**
19	C-LEACH	X	X	X	√	√		X	X	X	X	X	**X**	**X**
20	SOJK	√	√	√	√	√	√	√	√	√	√	√	√	√

provides the API required for providing the status of a device or transmitting the data routed from a device through the controller to a cloud through either a Gateway or through a web service server. Both the WEB server and the gateway connected to the Internet on to which the cloud is interfaced. The remote users are connected to the cloud or the restful server through the Internet. The prototype network is simple, mostly connected using a peer to peer or a parallel connection except that the devices are connected through a Cluster.

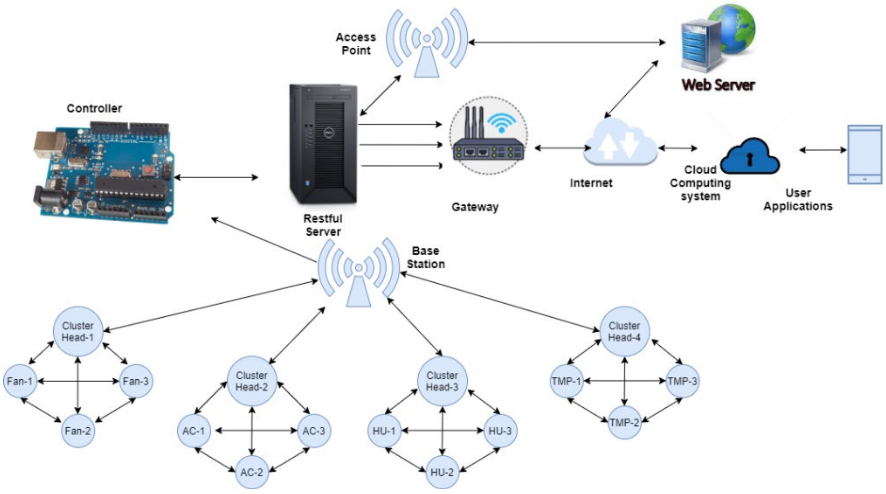

Fig. 1. Typical prototype network.

This model is used as an experimental model for implementing the changes proposed and showing how the performance of the IoT network improves. Every cluster possesses three inputs. There are three such clusters used in the networks all communicate using the Wi-Fi standard protocol, which cannot be the case in reality.

One can see from the figure that different techniques are used within the IoT network, which includes sensing, controlling, service rending, and heterogeneous communication, storing and retrieving the data from databases for building an IoT network. The details of the equipment used for networking shown in Table 2.

Table 2. Node identification within the prototype IoT network.

No.	Device Number	Device Name	No.	Device Number	Device Name
1	N0010	Fan-1	**13**	N0040	TEMP-1
2	N0011	Fan-2	**14**	N0041	TEMP-2
3	N0012	Fan-3	**15**	N0042	TEMP-3
4	CLUS1	Cluster Head-1	**16**	CLUS4	Cluster Head-4
5	N0020	AC-1	**17**	N005	Controller
6	N0021	AC-2	**18**	N006	Restful server
7	N0022	AC-3	**19**	N007	Gateway
8	CLUS2	Cluster Head-2	**20**	N008	Access point
9	N0030	HU-1	**21**	N009	WEB server
10	N0031	HU-2	**22**	N00A	Cloud
11	N0032	AC-3	**23**	N00B	User Device
12	CLUS3	Cluster Head-3			

4.1. Experimenting model

Time consumed by every device and within every Layer for sensing, processing, protocol conversion, receiving, and transmitting has to be computed and added to find the overall response time. Computing the response time is the real challenge before we even think of improving the performance of the IoT network. A process is added into every Layer to log the performance related data like the time at which the data is received, time taken to process the data, and time taken to transmit the data. The time taken by the process to compute the response time needs to be subtracted from the total time to arrive at the actual time taken to handle the data within a layer.

$$totL_i = tp_{Li} + ttL_i + tcLi + ttcL_i + t_{RLi} + t_{PLi} + t_{TLi} \quad (1)$$

where tp_{Li} = Time taken to process the Data in ith Layer, ttL_i =Time Taken to Transmit the data, tcLi = Time Taken to convert the data the Data in converter, $ttcL_i$ = Time Taken to Transmit the data from the converter, t_{RLi} = Time taken to receive the data at i+1th Layer, t_{PLi} = Time taken to process the data at i+1th Layer after receiving the data, $t_{TLi} = t_{SLi}$ = Time take to store the computations in a remote server, $totL_i$ = Total Time Taken for transmission and reception of the data

TT - Total Time Taken for transmitting the data from one end of IoT network to another end or vice versa.

$$TT = \sum_1^7 totLi \quad (2)$$

The data captured at each of the Layer is stored in a database in the format shown in Table 3, and Fig. 2 shows the Data Logging process used for experimentation and processing the response time.

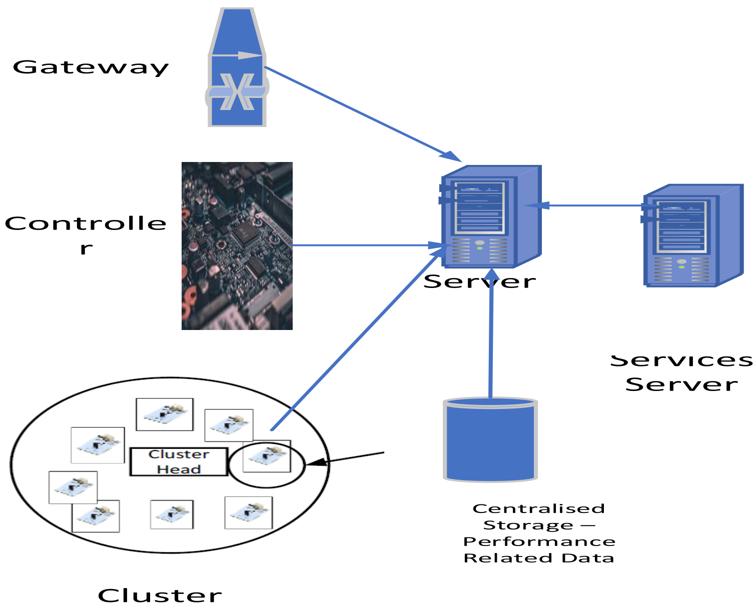

Fig. 2. Data logging process for computing the response time.

Table 3. Format of data structure for storing the data in the database.

Serial number of the field	Name of the Field	Type of Field
1	Layer Number	NUM
2	Device Code	CHAR
3	Time in Microseconds for Receiving the data	NUM
4	Time in Microseconds for Processing the data	NUM
5	Time in Microseconds for Transmitting the data	NUM
6	Time in Microseconds for Logging data in the remote server	NUM

4.2. Performance computations – Prototype model

The performance computations of the prototype model have been made , considering three data packets that Move across the devices which are situated in different layers. The system is made to work, and the response time computations are collected and logged into a remote server. Four clusters are used to sense and transmit the data across the network. In each of the clusters, three devices are connected using three paths emanating from any of the devices. Every device can communicate using any of the paths as long as the link is working. Wifi based communication is used as a mode of communication, considering the speed of 11mbps. The issue of heterogeneity has not been considered in the network. Power depletion of 0.001 WATS for every 36 bytes of transmission is considered. The time taken to process the data and to log the data at every Layer has been computed and logged into a database. Table 4 shows computational details from the transmission side, and Table 5 shows the computational details from the conversion and reception side.

Data summarization has been done considering different parameters that affect the performance of the prototype model. The parameters considered include Power reserve within the devices, size of the data transmitted, the total power consumed for transmission, conversion and reception of the data, Amount of power depletion after completion of the data transmission, estimation of the number of packets that would be transmitted by the time devices goes to the zero states, number of paths used for effecting the Transmission and Total time taken to transmit the data in Microseconds is shown in Table 4.

Table 4. Summarisation of the performance computations for the prototype model.

Packet Number	Original total power of the devices in the clusters	Size of the data Transmitted	Total Power Consumed	Power Depletion	Estimated Packets transmitted before the power goes to 0 stage	Number of Paths used for transmission	Number of conversions used	Response Time
1	48.000	1204	47.971	0.029	1655	48	0	875.78
2	47.971	1312	47.954	0.017	2822	48	0	954.33
3	47.954	1420	47.936	0.018	2664	48	0	1032.87

Data Analysis has been carried to find the Behavior of the networks as the parameters vary. Data Analysis carried include Behavior of the response time Vs. The number of communication paths (Fig. 3), Response time Vs. Size of the data transmitted (Fig. 4), Response time Vs. The number of packets transmitted at zero power state (Fig. 5), Power depletion Vs. Size of the data transmitted (Fig. 6), Total power consumed Vs. The number of packets Transmitted (Fig. 7).

It is observed from the figures that power depletion takes place quite drastically as the size of the data to be transmitted increases drastically; there is no effect on the response time when the number of paths used for transmission of the packets remains the same. Response time increases proportionally to the size of the data transmitted; total power consumed reduces drastically as the size of the data to be transmitted increases.

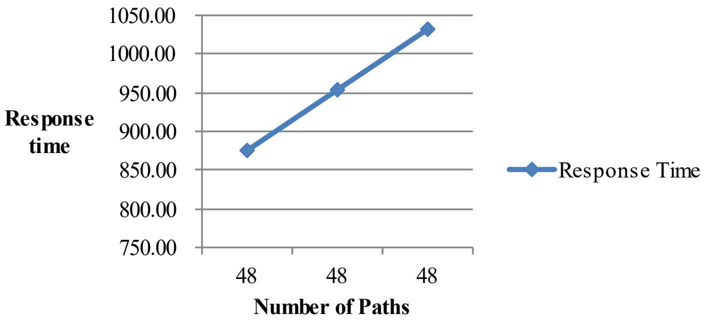

Fig. 3. Response time vs. number of paths.

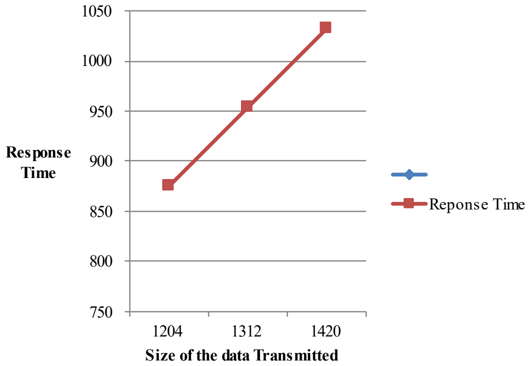

Fig. 4. Response time vs. size of the data transmitted.

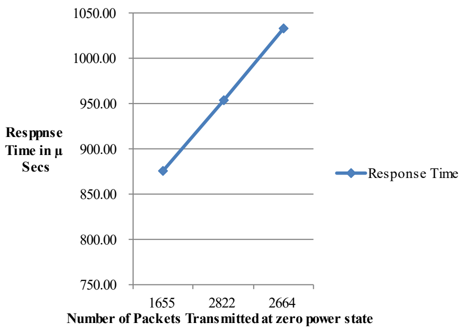

Fig. 5. Response time vs. number of packets transmitted at zero power state.

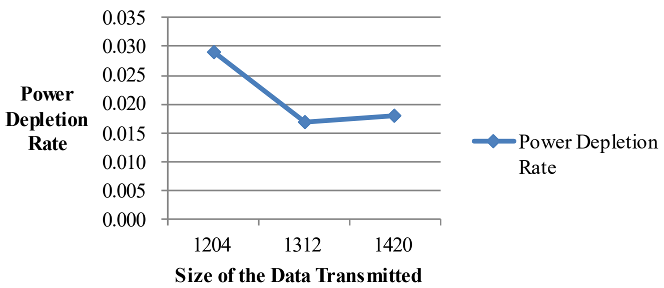

Fig. 6. Power depletion vs. size of the data transmitted.

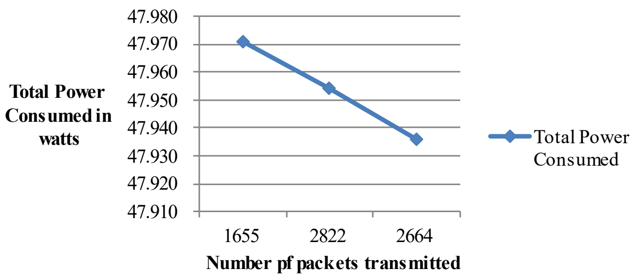

Fig. 7. Total power consumed vs. number of packets transmitted.

5. Investigations and Findings

5.1. Networking of devices in clusters for enhancing the fault tolerance through the creation of alternate paths for communication.

The networking of the devices in a cluster is the key to improve the performance of every cluster. Multi cluster-based communication also helps in improving the response time as many the number of paths existing in the network improves. The network used for connecting the devices in the cluster can be established using many topologies such as crossbar, butterfly, multi-stage, etc. In this paper, the crossbar based networking is considered to find the extent to which performance improvement can be achieved through an increase of paths for transmitting the data. The network topology is shown in Fig. 8.

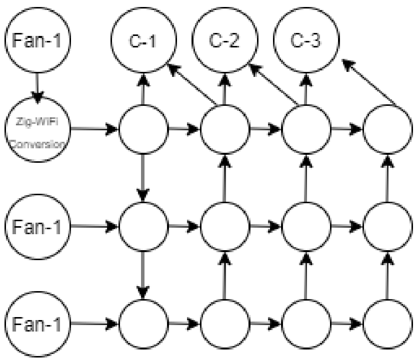

Fig. 8. Crossbar based networking topology.

A crossbar network topology deals with N Inputs (Fault rates of the Incoming Devices) and M Outputs (Fault rates of the Outgoing devices, there is one switch Box associated with each input and output, which is an additional hardware element added into the network. The switch box in row I and column j are responsible for connecting the network input on row I to the network output on row i. The box, as such, is called ij Box.

Each switch box Forwards the data received from its left link to its right link, which means propagate the data horizontally and also forward the data flow through its bottom link to its top link. The switch box also is capable of moving data from its left lint to the Top Link. Every link at most can carry only one data element, and each switch box will be able to process two data elements at the same time. A switch box can forward data from its left link to its right link while at the same time forward the data from its bottom Link to Top Link.

If input 3 is to be sent to output 5, the data will be received by switch box (3,1), which will forward it to (3,2) and so on until it reaches (3,5), which will then foreword it to (2,5) and then to (1,5) which will then be sent to the connected device. The routing strategy is rather obvious. For example, if we want to send a message from input 3 to output 5, we will proceed as follows. The input will first arrive to switch box (3, 1), which will forward it to (3, 2) and so on until it reaches switch box (3, 5). This switch box will turn the message into column 5 and forward it to the box (2, 5), which will send it to the box (1, 5), which will send it to its destination.

From this network, one can see that any input-output combination can be realized as long as there is no collision at the output (No two inputs are competing for the same output line). This network thus is quite suitable when the process is quite faster and just involves transmission such as the transmission of data from a sensor. The connectivity of the crossbar can be analysed to assess the failure rates of the individual components.

q_l = probability that a Link is Faulty
$1-q_i$ = probability that a Link and the switch box is not Faulty

Counting from 1, for input I to be connectable to output j, we have to go through a total of $I + j$ links. The probability that all of them are fault free is P_l^{i+j}

The probability that a network will be fault-free =

$$Q = \sum_{i=1}^{N} \sum_{j=1}^{M} p_l^{i+j} = p_l^2 \frac{1-p_l^N}{p_l} \frac{1-p_l^M}{p_l} \qquad (3)$$

It has been proved that the reliability of the device-based clusters tremendously improves through building networks like crossbar, Butter Fly, etc.

The number of communication paths from any node can be computed using Eq. (4), where n = Number of Inputs, if n = 4, then the number of paths available for communication from any node is = 16. The number of paths for communication thus increases with the adaption of network topology like crossbar.

$$\text{Number of paths} = 2^{n+1} \qquad (4)$$

5.2. Handling Heterogeneity among the devices

Heterogeneity is the main issue when it comes to IoT networks. Heterogeneity can be handled through a Hardware converter or using the devices that have an interface to the communication protocols among which the conversion is needed. In Fig. 9, it has been shown that FAN1 communicates in ZigBee protocol and the output transmitted using this protocol needs to be converted to Wi-Fi before the same is transmitted to the network for communication. Conversion consumes time for pocketing and de-pocketing and also to communicate to the next Layer. Processing time will also be consumed for storing the results into a remote database.

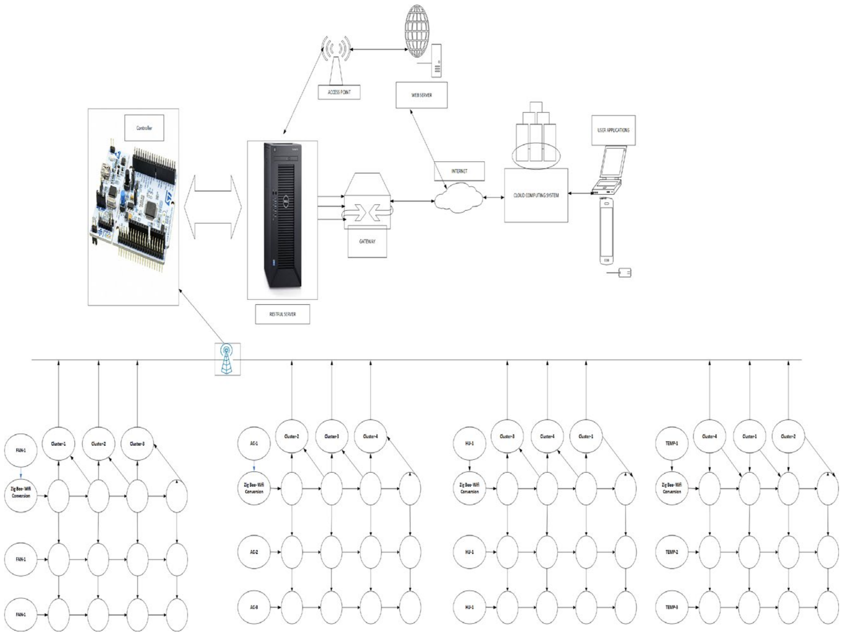

Fig. 9. Modified IoT network based on the investigations presented in Sections 5.5. 5.6 and 5.7.

5.3. An efficient algorithm for power maximisation for increasing longevity (Modified LEACH)

Many algorithms have been reviewed in Section 2.0, and the parameters used for selecting the cluster head have been presented. Most of the algorithms concentrated on the minimization of the power consumed for communication through a selection of a cluster head. Every device will be configured as a cluster head, and the same is used not only for sensing but also for communicating the data to the cluster head. The sensors thus have a load of not only sensing but also transmitting when selected as cluster head. The number of paths that can be used for routing the data to a cluster head is fixed. There will be a complete setback when the designated cluster head fails for any reason. To avoid this bottleneck, the availability of several clusters' heads and the selection of one of the cluster heads based on its working status will improve the performance of the cluster head drastically. This also frees the sensing devices from the overhead of communicating with the base station. In a network, there can be several clusters, and all the clusters can share a set of cluster heads, thereby reducing the number of cluster heads required form communication.

5.4. Algorithmic steps

SOJK algorithm is shown in Table 5, which shows the algorithmic steps implemented within each of the sensor devices.

Table 5. SOJK Algorithmic steps.

Step	Type of Execution Done
1.	Deploy the Switches
2.	Deploy Cluster Heads
3.	Maintain a repository containing the details of the Cluster Heads (CRP = Cluster Rated power, CLP= Cluster Leftover Power, CNOP = Number of Packets dispatched, CSDATA = size of the data despatched to the base station
4.	Maintain connectivity of cluster heads to the topmost switches
5.	Deploy the Sensors and switches that connect the sensors to the clusters
6.	Maintain a repository containing the details of the Sensors (SRP = Sensor Rated power, SLP= Sensor Leftover Power, SNOP = Number of Packets dispatched to a chosen cluster head, SSDATA = size of the data despatched to cluster head from the sensor
7.	Maintain a repository of paths from the sensor through the switches till a cluster head and find the minimum power of the path as shown in Table 6
8.	Select the paths in ascending order of the power that are required to transmit the data in parallel channels. The number of channels required to transmit the data depends on the size of the data to be sent.
9.	Break the data into the number of the channel selected and form into packets as per the protocol supported by the device
10.	Transmit the Packets through the channels selected
11.	Every device updates its status after transmission of the data is completed
12.	Repeat the steps 8-12 every time a need arises to transmit the data to the remote cloud. Or when data is to be received from the cloud

Table 6. Path layout table with minimum power in the path from Sensor F1.

DEV-1	DEV-2	DEV-3	DEV-4	DEV-5	DEV-6	DEV-7	DEV-8	DEV-9	Minimum of the Left over-power
SW1	Cluster-1								2.99
SW1	SW2	Cluster-2							2.98
SW1	SW2	SW3	Cluster-3						2.66
SW1	SW2	SW3	SW4	Cluster-4					2.45
SW1	SW5	SW6	SW2	Cluster-2					2.98
SW1	SW5	SW6	SW7	SW3	Cluster-3				2.99
SW1	SW5	SW6	SW7	SW8	SW4	Cluster-3			2.45
SW1	SW5	SW9	SW10	SW6	SW2	Cluster-1			2.78
SW1	SW5	SW9	SW10	SW11	SW7	SW3			2.99
SW1	SW5	SW9	SW10	SW11	SW12	SW8	SW4	Cluster-3	2.88

5.5. The topology of modified IoT network

The prototype IoT network has been modified to cater to crossbar networks in the devices layer and inclusion of the clustering algorithm presented in section 5.6. The Modified Network is shown in Fig. 9. In the revised network, crossbar networking topology has been introduced that considers four clusters, each supporting three inputs and four sharable cluster heads for effecting communication to the base station.

5.6. Overall performance computations of the modified network

The performance computations of the modified network have been carried. considering three data packets that move across the devices which are situated in different layers. The system is made to work, and the response time computations are collected and logged into a remote server. Four clusters are used to sense and transmit the data across the network. In each of the clusters, three devices are connected using 15 paths emanating from any of the devices.

Every device can communicate using any of the paths as long as the link is working. Wi-Fi based communication is used as a mode of communication, considering the speed of 11mbps. The issue of heterogeneity has been considered in the network through the conversion of ZigBee packets to Wi-Fi packets, which is the main protocol used for effecting the communication. Power depletion of 0.001 WATS for every 36 bytes of transmission is considered. The time taken to process the data and log the data at every Layer has been computed and logged into a database.

Data summarization has been done considering different parameters that affect the performance of the prototype model. The parameters considered include Power reserve within the devices, size of the data transmitted, Total power consumed for transmission, conversion and reception of the data, Amount of power depletion after completion of the data transmission, estimation of the number of packets that would be transmitted by the time devices goes to the zero state, number of paths used for effecting the Transmission and Total time taken to transmit the data in Microseconds is shown in Table 7.

Table 7. Data summarisation of revised IoT model.

Packet Number	Original total power of the devices in the clusters	Size of the data Transmitted	Total Power Consumed	Power Gained due to cluster Head rotation	Actual Power Consumed	Power Depletion	Estimated Packets transmitted before the	Number of Paths used for transmission	Number of conversions used for Handling Heterogeneity	Response Time
1	48.000	1279	47.971	0.015	47.986	0.014	3429	372	4	582.089
2	47.971	1387	47.954	0.015	47.939	0.032	1499	372	4	690.877
3	47.954	1480	47.936	0.015	47.921	0.033	1453	372	4	844.914

Data Analysis has been carried to find the Behavior of the networks as the parameters vary. Data Analysis carried include Behavior of the response time Vs. The number of communication paths (Fig. 10), Response time Vs. Size of the data transmitted (Fig. 11), Response time vs. The number of packets transmitted at zero power state (Fig. 12), Power depletion Vs. Size of the data transmitted (Fig. 13), Total power consumed Vs. Several packets Transmitted (Fig. 14). These figures show the comparative analysis of the revised model about the prototype model.

It is observed from the figures that the performance of the revised IoT model is much better compared to the prototype model. Performance Improvement of 15% has been achieved while making the IoT network fault Tolerant. It has also been observed that power depletion takes place quite drastically as the size of the data to be transmitted increases drastically; there is no effect on the response time when the number of paths used for transmission of the packets remains the same. Response time increases proportionally to the size of the data transmitted; total power consumed reduces drastically as the size of the data to be transmitted increases.

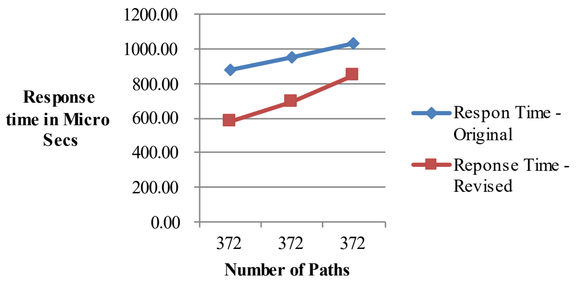

Fig. 10. Comparative analysis of response time vs., number of communication paths.

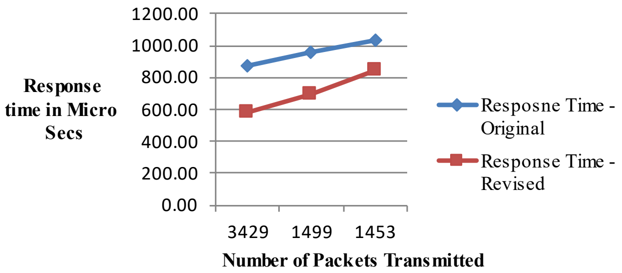

Fig. 11. Comparative analysis of response time vs. number of packets transmitted at zero power state.

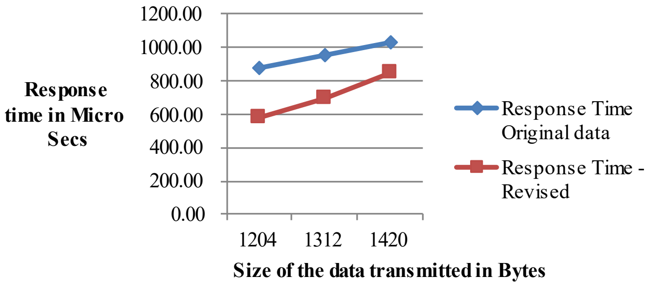

Fig. 12. Comparative analysis of response time vs. size of the data transmitted.

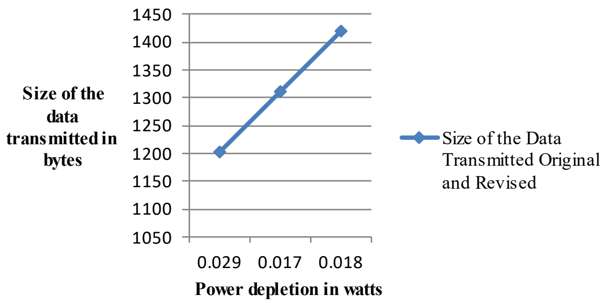

Fig. 13. Comparative analysis of power depletion vs. size of the data transmitted.

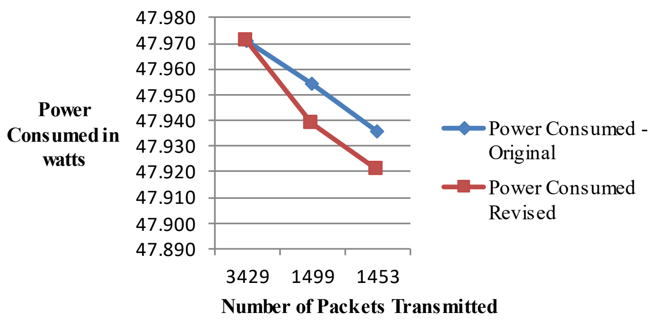

Fig. 14. Comparative analysis of total power consumed vs. number of packets transmitted.

5.7. Comparative Analysis of the performance of IoT networks

The SOJK Model proposed in this paper has been compared with LEACH, ILEACH, and RLEACH. Comparison is shown over several parameters that are computed on the Average basis. The Parameters considered include the average size of the data in bytes Transmitted in three packets, Average power Depletion for three packets of data Transmission in watts, Average number of the estimated number of packets transmitted at Zero power state, Average number of paths used for transmission, number of conversions used and Average Response time in Micro Seconds. The Comparison is shown in Table 8. The Comparison clearly shows that the SOJK model performs 35% more than any other nearest model.

Table 8. Comparative analysis of clustering models.

Parameter used	LEACH	ILEACH	RLEACH	SOJK LEACH
The average size of the data in bytes Transmitted in three packets	1312	1312	1312	1312
Average power Depletion for three packets of data Transmission in watts	0.432	0.322	0.271	0.021
The average number of Estimated number of packets transmitted at Zero power state	1123	1275	1279	2127
The average number of paths used for transmission	15	15	15	372
Number of conversions used	0	0	0	4
Average Response time in Micro Seconds	954.328	901.77	895.38	705.96

6. Conclusions

IoT networks are affected in performance continuously due to the existence of heterogeneity, failure of communication paths, too much traffic in a specific channel, fast depletion of power in the sensing devices, and huge generation of data in minimum time. Clustering mechanisms have been tried to address the issue of power depletion, and never the issue of heterogeneity addressed.

Communication is affected considering a fixed number of paths, generally a single path from a sensor to a communicating device. The latencies that exist in each of the Layer have also been addressed. The power depletion of sensors gets reduced when the sensors are not used for transmission and that when the transmission is done through separate cluster heads on a sharable basis.

Maximum data can also be transmitted using several communication paths, thereby improving the response time of the IoT networks. Crossbar network provides several communication paths and provides several alternative paths that help not only in improving the performance but also makes the network fault tolerant.

The performance of the revised IoT network improved by 15% when compared to the performance of the prototype network due to implementation of SOJKLEACH Algorithm and implementation of crossbar networking topology for implementing device level network. The SOJKLEACH algorithm performed 35% better than other clustering algorithms when it comes to response time.

Performance improvements have also been witnessed considering the number of data packets dispatched before the devices become dormant, an improvement from 1279 to 2127 packets of data. The number paths available for transmission increased from 15 to 372 due to introduction of crossbar network in the device layer. The percentage power depletion on transmission of data packets improved from 0.271 to 0.021.

7. Future Scope

In this paper performance enhancement of IoT network is addressed just focussing on the device level. Further research is in progress focussing on performance improvement considering other layers existing in an IoT network.

References

1. Boyinbode, O.; Le, H.; Mbogho, A.; Takizawa, M.; and Poliah, R. (2010).A survey on clustering algorithms for wireless sensor networks. *Proceedings of* 2010 13*th International Conference on Network-Based Information Systems*, Takayama, Japan, 358-364.
2. Kim, J.; and Lee, J. (2012). Cluster-based mobility supporting WMN for IoT networks. *Proceedings of the* 2012 *IEEE International Conference on Green Computing and Communications*, Besancon, France, 700-703.
3. Tao, X.; and Ji, C. (2014).Clustering massive small data for IoT. *Proceedings of the* 2014 2*nd International Conference on Systems and Informatics* (*ICSAI* 2014), Shanghai, China, 974-978.
4. Younis, O.; Krunz, M.; and Ramasubramanian, S. (2006). Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network*, 20(3),20-25.
5. Azad, P.; Sharma, V. (2013). Cluster head selection in wireless sensor networks under fuzzy environment. *International Scholarly Research Notices*, Volume 2013 |Article ID 909086, 1-8.

6. Bhupathi.; and Sastry, J.K.R. (2019). A framework for effecting fault tolerance within IoT network. *Journal of Advanced Research in Dynamical & Control Systems*, 10(2) Special Issue, 424-432.
7. Khediri, S.E.L.; Nasri, N.; Wei, A.; and Kachouri, A. (2014). A new approach for clustering in wireless sensors networks based on LEACH. *Procedia Computer Science*, 32, 1180-1185.
8. Mamta (2014). Various clustering techniques in wireless sensor network. *International Journal of Computer Applications Technology and Research*, 3(6), 381-384.
9. Singh, S.P; and Sharma, S.C. (2015). A survey on cluster-based routing protocols in wireless sensor networks. *Procedia Computer Science*, 45, 687-695.
10. Wang, Z.; Wu, R.; Sa, Q.; Li, J.; Fan, Y.; Xu, W.; and Zhao, Y. (2016). An improved cluster routing structure of IoT. *International Conference on Communications*, *Information Management and Network Security* (*CIMNS* 2016), 326-328.
11. Mahajan, S.; and Dhiman, P.K. (2016). Clustering in wireless sensor networks: A review. *International Journal of Advanced Research in Computer Science*, 7(3), 198-201.
12. Kwon, M.; and Park, H. (2016). The cluster formation strategies for approximate decoding in IoT networks. *Proceedings of the International Conference on Information Networking* (*ICOIN*), Kota Kinabalu, Malaysia, 366-368.
13. Xu, L.; Collier, R.; and O'Hare, G.M.P. (2017). A survey of clustering techniques in WSNs and consideration of the challenges of applying such to 5G IoT scenarios. *IEEE Internet of Things Journal*, 4(5), 1229-1249.
14. Puschmann, D.; Barnaghi, P.; and Tafazolli, R. (2017). Adaptive clustering for dynamic IoT data streams. *IEEE Internet of Things Journal*, 4(1), 64-74.
15. Ju, Q.; and Zhang, Y. (2017). Adaptive clustering for the internet of battery-less things. *Proceedings of the IEEE Wireless Communications and Networking Conference* (*WCNC*). *San Francisco*, *CA*, 1-6.
16. Ananthi, V.J.; Chinnalagi, V.; Murugeshwari, R.; Priyadharshni, T.; and Rajyalakshmi, K. (2017). An effective performance of smart sensor network using IoT. *International journal of Advance Research*, *Ideas*, *and Innovations in Technology*, 3(2), 638-646.
17. Fouladlou, M.; and Khademzadeh, A. (2017). An energy-efficient clustering algorithm for wireless sensor devices in the Internet of Things. *Proceedings of the Artificial Intelligence and Robotics* (*IRANOPEN*). *Qazvin*, 39-44.
18. Bhandari, S.; Sharma, S.K.; and Wang, X. (2017). Cloud-assisted device clustering for lifetime prolongation in wireless IoT networks. *Proceedings of the IEEE* 30*th Canadian Conference on Electrical and Computer Engineering* (*CCECE*), Windsor, ON, 1-4.
19. Choi, D.-K.; Jung, J.-W.; Kang, H.-W.; and Koh, S.-J. (2017). Cluster-based CoAP for message queueing in Internet-of-Things networks. *Proceedings of the* 19*th International Conference on Advanced Communication Technology* (*ICACT*), *Bongpyeong*, 584-588.

20. Ashwini, M.; and Rakesh, N. (2017). Enhancement and performance analysis of LEACH algorithm in IoT. *Proceedings of the International Conference on Inventive Systems and Control* (*ICISC*), *Coimbatore*, Inida, 1-5.
21. Al-Janabi, T.A.; and Al-Raweshidy, H.S. (2017). Optimised clustering algorithm-based centralized architecture for load balancing in IoT network. *Proceedings of the International Symposium on Wireless Communication Systems* (*ISWCS*). *Bologna*, 269-274.
22. Sung, Y.; Lee, S.; and Lee, M. (2018). A multi-hop clustering mechanism for scalable IoT networks. *Sensors*, 18(4), 961.
23. Nguyen, T.M.C.; Hoang, D.B.; and Dang, T.D. (2018). A software-defined model for IoT clusters: Enabling applications on-demand. *Proceedings of the International Conference on Information Networking* (*ICOIN*), *Chiang Mai*, 776-781.
24. Wang, Z.; Qin, X.; and Liu, B. (2018). An energy-efficient clustering routing algorithm for WSN-assisted IoT. *IEEE Wireless Communications and Networking Conference* (*WCNC*), *Barcelona*, Spain, 1-6.
25. Kumar, J.S.; and Zaveri, M.A. (2018). Clustering approaches for pragmatic two-layer IoT architecture. *Wireless Communications and Mobile Computing*, Volume 2018, Article ID 8739203 1-16.
26. Lin, K.; Wang, D.; Xia, F.; and Ge, H. (2018). Device clustering algorithm based on multimodal data correlation in cognitive internet of things. *IEEE Internet of Things Journal*, 5(4), 2263-2271.
27. Behera, T.M.; Samal, U.C.; and Mohapatra, S.K. (2018). Energy-efficient modified LEACH protocol for IoT application. *IET Wireless Sensor Systems*, 8(5), 223-228.
28. Choi, D.-K.; Jung, J.-W.; and Koh, S.-J. (2018). Enhanced cluster-based CoAP in internet-of-things networks. *International Conference on Information Networking* (*ICOIN*), *Chiang Mai*, 652-656.
29. Behera, T.M.; Mohapatra, S.K.; Samal, U.C.; Khan, M.S.; Daneshmand, M.; Gandomi, A.H. (2019). Residual energy based cluster-head selection in WSNs for IoT application. *IEEE Transactions on Internet of things*, 6(3), 5132-5139
30. Sastry, J.K.R.; and Bhupathi. (2020). Enhancing fault tolerance of IoT networks within device layer. *International Journal of Emerging Trends in Engineering Research*, 8(2), 491-509.
31. Reddy, G.A.; Upendra, Y.; Sastry, J.K.R.; and Bhupathi. (2020). An approach to compute fault tolerance of an IoT network having clustered devices using crossbar networks. *International Journal of Emerging Trends in Engineering Research*, 8(4), 987-1004.
32. Anjana, A.; Chand, G.G.; Kiran, S.K.; Sastry, J.K.R.; and Bhupathi. (2020). On improving fault tolerance of IoT networks through butterfly networks implemented at services layer. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(2), 2096-2115
33. Bhanu, J.S.; Sastry, J.K.R.; Kumar, P.V.S.; Sai, B.V.; and Sowmya, K.V. (2019). Enhancing performance of IoT networks through high-performance computing. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(3), 432-440.

34. Sowmya, K.V.; and Sastry, J.K.R. (2018). Performance evaluation of IoT systems – basic issues. *International Journal of Engineering & Technology*, 7(2.7), 131-137.
35. Sastry, J.K.R.; Ramya, G.S.; Niharika, V.M.; and Sowmya, K.V. (2020). Performance optimization of IoT networks within the gateway layer. *Recent Advances in Computer Science and Communications*, 13(6), 1338-1346.
36. Krishna, M.S.R.; Sastry, J.K.R.; and Bhanu, J.S. (2017). Building fault tolerance within wireless sensor networks: a butterfly model. *Research Journal of Applied Sciences*, 12(2), 139-147.
37. Priya, B.V.; and Sastry, J.K.R. (2018). A comparative analysis of the methods used for building information / content-centric networks over software-defined networks. *International Journal of Engineering & Technology*, 7(2.7), 946-753.
38. Pavithra, T.; Sastry, J.K.R. (2018). Strategies to handle heterogeneity prevalent within an IOT based network. *International Journal of Engineering & Technology*, 7(2.7), 77-83.