FAULT TOLERANCE FOR CONCURRENT DATA COLLECTION IN IOT NETWORKS

Project report submitted in partial fulfillment of the requirement for the
Degree of Bachelor of Technology
In
Computer Science and Engineering
By

Nancy (141206)
Vrinda (141208)

Under the supervision of
Sh. Arvind Kumar

To

Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology Waknaghat, Solan-173234,
Himachal Pradesh**

CERTFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Fault Tolerance for Concurrent Data Collection in IoT Networks "** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2015 to December 2015 under the supervision of **Sh. Arvind Kumar** (Assistant Professor Grade-II, Computer Science and Engineering Department).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Nancy, 141206.

Vrinda, 141208.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Sh. Arvind Kumar

Assistant Professor (Grade-II)

Computer Science and Engineering Department

Dated:

ACKNOWLEDGEMENT

We owe our profound gratitude to our project supervisor **Sh. Arvind Kumar**, who took keen interest and guided us all along in my project work titled **—Fault Tolerance for Concurrent Data Collection in IoT Networks**, till the completion of our project by providing all the necessary information for developing the project. The project development helped us in research and we got to know a lot of new things in our domain. We are really thankful to him.

TABLE OF CONTENTS

Title **Page Number**

1. INTRODUCTION

2. LITEATURE SURVEY

3. SYSTEM DEVLOPMENT

4. PERFORMANCE ANALYSIS

5. CONCLUSION

LIST OF FIGURES

1. INTRODUCTION

1.1 The Internet of Things

1.1.1 About IoT

“The Internet of Things (stylized Internet of Things or IoT) is the cyber web working of physical gadgets, vehicles, structures and different things- embedded with electronics, software , sensors, activators and system network that empower these articles to gather and trade information.” In 2013 the global standards initiative on Internet of Things (IoT-GSI) characterized the IoT because the origin of your info network. IoT is largely uniting all of the devices of our daily living to cyber web as a way to carry out our living easier. “Things,” in IoT refers to every device from lights to vehicles etc. of our day to day life. These things are connected to a network so as to simplify our work life.

Researchers looks “Things” as an “inseparable blend of hardware, software, data and service”. These devices gather information from the nearby environment.

IoT is the leading research subject as nowadays focus is shifting towards ubiquitous computing from traditional computing.

IoT covers different aspects like transportation, construction, medical sciences, home appliances, shopping experiences etc.

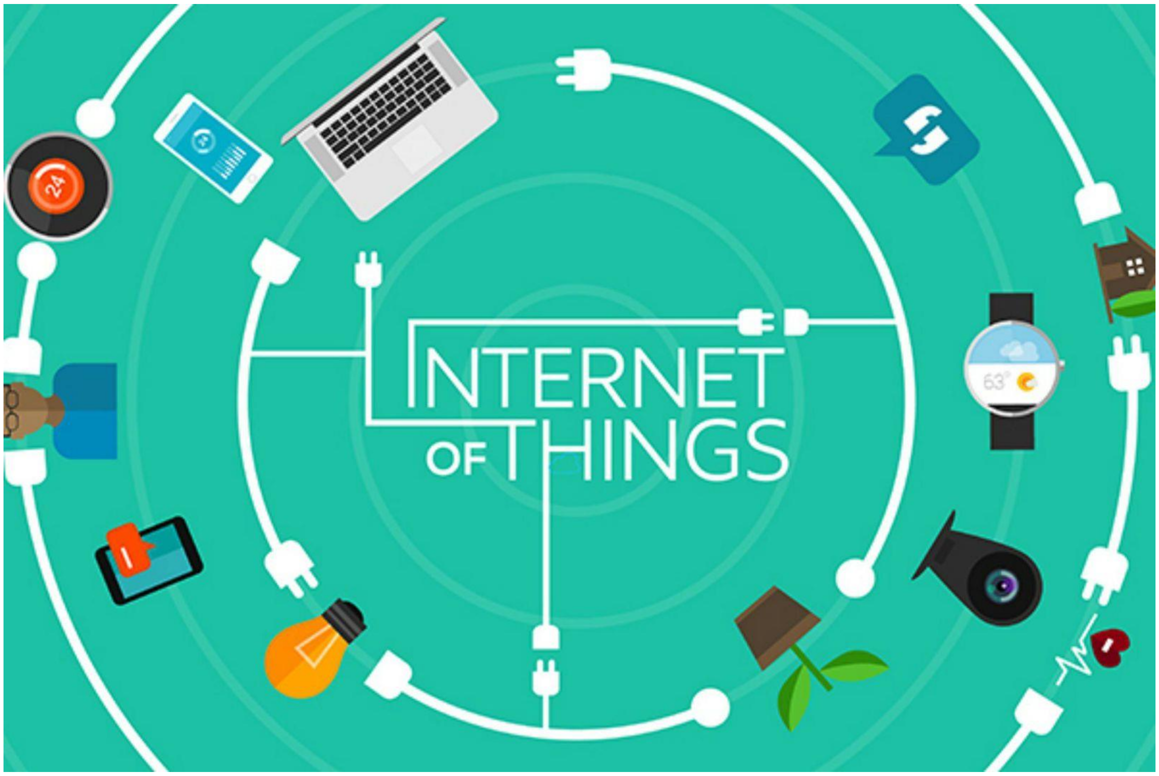

Fig.1.1.1 Application of IoT

1.1.2 Growth of the IoT

IoT is one of the main research point as it is new and a huge no of individuals around 87% have even not known about it. Be that as it may, IoT has been a major part of our life from long as ATMs which go back to 1980s. It is anticipated that around Billion articles will be associated with the web by 2020 out of which around 250 million will be vehicles. Indeed, even the market of brilliant watches has developed radically in the previous couple of years.

So it is clear that IoT is sought after and the gadgets associated with the web are developing to expand drastically.

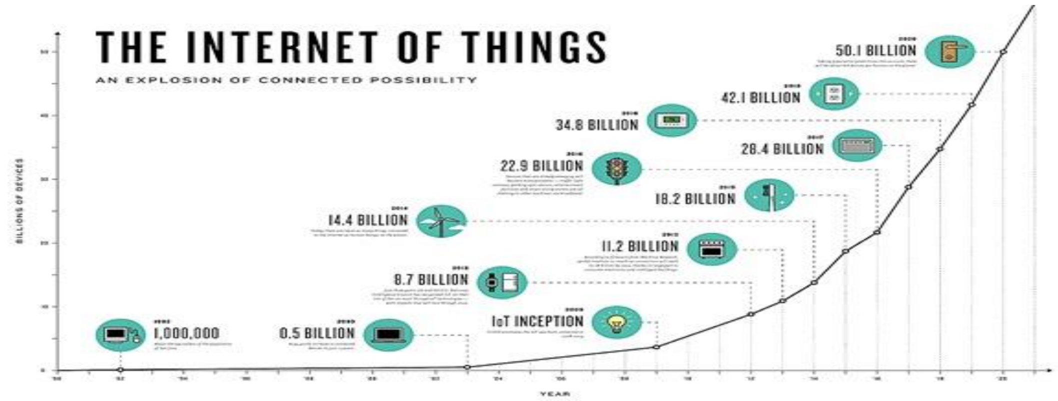

Figure 1.2 Growth of IoT

1.1.3 Long time to value

IoT activities can take a long time. From business case change or advancement to check of thought for full-scale rollout, each period of the system can be loaded down with troubles. In light of our work with customers, we recommend that associations take after a five-organize system to restrain the time required to pass on their IoT wanders, while intensifying the arrival on these exercises.

There are numerous projects accessible that encourages new associations to help them for the improvement of different IoT ventures. Indeed, even the legislature is concentrating on IoT by actualizing new plans like Smart Cities, free Wi-Fi Zones and so on.

1.1.4 Challenges

In this report we discuss about Fault tolerance whose identification and avoidance is a major challenge. As we all know that this particular fault changes the rank of a particular node, thus making it change the topology of the network. This fault if occurred in the base station node can cause a major challenge for the user.

1.2 Problem Statement

Implementation and detection of faulty node be it a cluster member, cluster head or base station and analysis of change in topology of the network structure.

1.3 Objective

The aim is to analyze and detect a fault in a network structure which uses concurrent data transmission and its effect on delay.

1.4 Methodology

In this project we simulate a network structure on the Linux based simulator COOJA. Our prime methodology is to detect a node which makes the network structure faulty. Henceforth, making a fault tolerant network structure.

- **Fault Tolerance Algorithm:** This step involves the development of a suitable combination of a fault tolerant algorithms that best serves our purpose. The combination can be then tested on various parameters such as delay, height of tree etc.

- **System Design:** We then create a proposed network structure on the network simulator i.e. COOJA. COOJA simulator is a network simulator specifically designed for Wireless Sensor Networks.

- **Deployment:** In this step we finally deploy our network design.

1.5 Organization of Project Report

In Chapter 1 we have discussed about IoT basics, the current growth in this field, the common challenges being faced by persons in implementing the IoT structures.

In Chapter 2 we will be discussing the basic terminology mentioned in a topic. We will be providing with facts and figures about different concepts we studied about those terms in different research papers.

In Chapter 3 we are going to provide a model of how the project is done on the basis analytical developments.

In Chapter 4 we have given a proper analysis of fault tolerance in concurrent data transmission.

In Chapter 5 we have provided with the conclusion that we derive from our project.

2. LITERATURE SURVEY

A writing audit intends to assess and translate all accessible research significant to a specific research question, or region. Its principle point is to show a reasonable assessment of the exploration zone of enthusiasm by directing a thorough and auditable system. The primary motivation behind our writing survey is excessively locate the applicable writing about adaptation to non-critical failure in simultaneous information transmission and their system conventions with the end goal of foundation think about, condense the current work and distinguish the hole in momentum research.

2.1) Title: “A Delay-Aware Data Collection Network Structure for Wireless Sensor Networks” (2011)

Wireless sensor systems comprise of a vast quantities of remote sensor hubs to assemble data. The hubs are battery-fueled gadgets. WSNs calculations ought to have a structure which has least postponement and backings vitality protection. The benefits of WSNs are: High versatility, far reaching detecting scope, and adaptation to internal failure. The hubs are smaller, light-weighted, and battery-controlled that can be utilized anyplace. Because of this, they can be used close to the objectives important to do successful detecting. The information gathered will be mulled over for arrange procedures and after that arrival to the client who is on an uncommunicative site. Sensor hubs must preserve their deficient vitality by every last way and remain dynamic with a specific end goal to keep up the scope.

The vitality is moderated by utilizing the standard of grouping. The working of this standard can be depicted as a system which is partitioned into different groups. In each bunch, one of the sensor hubs is picked as a group head (CH) and remaining are called bunch members (CM). The group head will gather every one of the information from its bunch individuals straightforwardly or in a multihop way i.e. not specifically but rather starting with one hub then onto the next. By sorting out remote sensor hubs into groups. Vitality misfortune is diminished by diminishing the quantity of hubs associated with long separation transmission Hence it brings about lessen information combination on the nodes.

2.1.2 Proposed Network Structure

The proposed network and its architecture as shown in figure 1 contains following:

1.Base Station

2. Cluster Head

3.Cluster Member

4. Network size N=16

5. Rank of each node is represented by *k*.

6. Delay is represented by T.

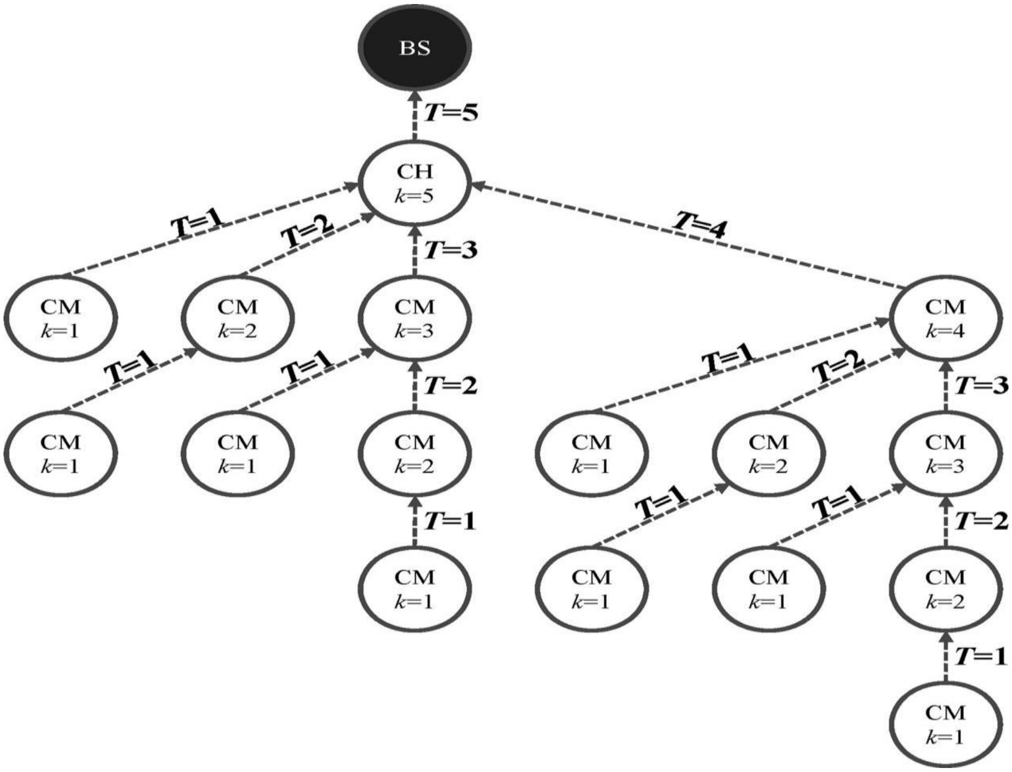

Figure 2.1.1: Proposed network architecture

1. Base Station: Short-go transceiver which interfaces a cordless telephone, PC, or different remote gadget to a focal center and enables association with a system.

2. Cluster Head: A node in a cluster that is responsible for collecting data from the sensors in its cluster and relay these data to the Base Station. The role of Cluster Head usually rotates between the nodes in cluster. It is the one with the highest rank in the network.

3. Cluster Member: A node in a cluster which is neither a base station nor a cluster head.

4. Network Size N: The network structure above is a tree structure to have maximum data collection the number of nodes N in the proposed network structure is limited to $N=2^p$ where p=1, 2, 3 …In the above diagram N=16.

5. Rank of Node *k*: Every node in the network is given a rank which lies between 1 and p. A node with rank *k* will form a data link with a node of a higher rank. This higher rank node will become the parent node of the node with the rank *k*.

6. Delay T: The delay of a network specifies how long it takes for a stream of data to travel across the network from one node or end point to another. There is a certain minimum level delay that will occur due to the time it takes to transmit a packet serially through a link.

2.1.3 Concept of Network Formation Algorithm

It has been demonstrated in the last area that the deferral in the information gathering procedure of a remote sensor system can be limited by utilizing the above system structure. Since vitality utilization is dependably a noteworthy issue in the investigation of remote sensor arranges, the target of the proposed organize development calculations is, in this way, to accomplish the proposed arrange structure while keeping the vitality utilization in the information accumulation process at low esteem.

A remote sensor hub can be considered as a gadget developed of three noteworthy units, to be specific the microcontroller unit (MCU), the handset unit (TCR), and the sensor board (SB). Every one of these units will devour a specific measure of vitality while working. The vitality devoured by a remote sensor hub can be communicated as

$$E_{i\_\mathrm{SN}} = E_{i\_\mathrm{MCU}} + E_{i\_\mathrm{TCR}} + E_{i\_\mathrm{SB}}$$

2.1.3.1 Network formation algorithm:

For systems with 2k hubs, where k=2, 3 ...the proposed arrange structure can be developed by the accompanying calculation.

1) The calculation thinks about the whole system as completely associated. In this paper, the term associated suggests there exists an information connect between two remote sensor hubs which transmit information parcels amid information gathering process. Two remote sensor hubs are characterized as separated from each other if there does not exist any immediate information connect between them. The association level of a remote sensor hub is demonstrates the quantity of information joins related with such hub. A hub with association level of 2 suggests that such a hub has shaped two information joins with three different hubs.

•For a system of N= 2k hubs, where K=2, 3 ...every hub will start with degree equivalent to N-1.

•The hubs will shape the set Ji additionally take another variable a, with the end goal that a=N/2.

2) After that we select a hubs from set J to shape set Js+1, with the end goal that ∑x,y € Ji+1z2xy is greatest. Here, zx y signifies the topographical separation between hub x and hub y. Remaining hubs from Ji will shape the set Ji+1. After that we will expel all information joins among hubs inside Js+1. At that point set the two iterators to values s = s+1 and a = a/2.

3) Repeat stage 2 until a < 2. Set g = 2.

4) Nodes with degree N-g frame set V. Hubs > N-g shape set T with the end goal that the quantity of hubs in the set V and set T are same. Associations among hubs in the two sets are lessened until the point that every hub in set V is just associated with a solitary hub in set T. Here, information joins are expelled by their separation. Points of interest of the enhancement technique are given in the later piece of this segment. In the wake of diminishing the quantity of associations, set g= g*2.

5) Repeat stage 4 until g=N.

The two hubs having a place with the last set T of stage 4 are having the most noteworthy association degree among the hubs in the system. These two hubs are, indeed, the hubs having a place with the set J in stage 2, when a = 1<2. Since these hubs are from the last set J produced from stage 2, they have not experienced the interconnection evacuation process. In this manner, these two hubs are interconnected with each other. Along these lines, the association degrees of these two hubs are constantly higher than the others. Subsequently, these two hubs are constantly incorporated into the set T in stage 4. Before the finish of stage 4, every one of these two hubs will have specifically associated youngster hubs with novel rankings, gave that the rankings of the kid hubs are lower than the two hubs. These two hubs are, in this manner, with association degrees equivalent to log (N).

2.1.5 Conclusion:

In this paper, a postponement mindful information accumulation organize structure and its development calculations are proposed. To cook for various applications, arrange development can be executed in either brought together or decentralized way. Two system development approaches are determined to give advanced outcomes to systems with various sizes. The execution of the proposed arrange structure is contrasted and a numerous group two-jump organize structure, a solitary chain organize structure, a base spreading over tree arrange structure, and a gathering tree arrange structure. The proposed arrange structure is appeared to be the most productive as far as information gathering time among all the system structures specified previously. The proposed arrange structure can extraordinarily lessen the information gathering time while keeping the aggregate correspondence separate and the system lifetime at adequate qualities.

2.2 Title: “Concurrent Data Collection Trees for IoT Applications”

This paper talks about the information gathering process for enormous volumes with a specific end goal to keep the general information accumulation process short. Web of Things (IoT) frameworks involve huge volumes of shrewd gadgets. Through trades of data, shrewd articles are equipped for thinking and create more elevated amount of insight. The viability of information accumulation forms is a key factor to the accomplishment of IoT frameworks as it can truly influence the freshness of the caught information. Effective information accumulation forms have been all around considered on tactile frameworks with static topologies and single information extraction point. It is protected to expect that for future IoT frameworks, an arrangement of sensors and middleware will be possessed and shared by numerous clients.

2.2.1 Characteristic of Concurrent Data Collection

- Concurrent information gathering trees are proposed to keep the general information accumulation length short. The proposed thought can extraordinarily lessen delays in simultaneous information accumulation forms.

- It includes considered gathering information from an extensive volume of people to a solitary information extraction point.

- In standard WSNs, sensor hubs are ordinarily claimed and overseen by a solitary gathering. In IoT applications, be that as it may, IoT gadgets can be mutually possessed by various clients or applications, who may trigger simultaneous information totals at the same time on a similar arrangement of hubs.

2.2.2 Overview of Concurrent Data Collection Trees

- Consider an IoT organize N = {n1, n2… n|N |} and an arrangement of base stations S = {s1, s2… s|S|}.

- It is accepted that all these |N | IoT hubs can speak with each other and achieve the base stations.

- Data gathered from various IoT gadgets are thought to be impeccably fusible, with the end goal that numerous got information parcels can be melded into one preceding sending to one's parent hub.

- Each simultaneous information conglomeration process will utilize an alternate base station (BS) to get to the IoT arrange and the aggregate number of simultaneous information streams is k.

- To keep up reasonableness among these clients, every single simultaneous datum stream should start and end in the meantime space.

- Parallel information streams ought to use a similar number of hubs at each vacancies.

- To abbreviate the general information gathering process, every datum stream ought to use the greatest conceivable number of hubs at each vacancy.

- In a system N with k simultaneous information accumulation forms, to such an extent that $|N| \geq k$, the most extreme number of hubs that can be used in the first schedule vacancy is given by,

$$U_{max} = lNl/k$$

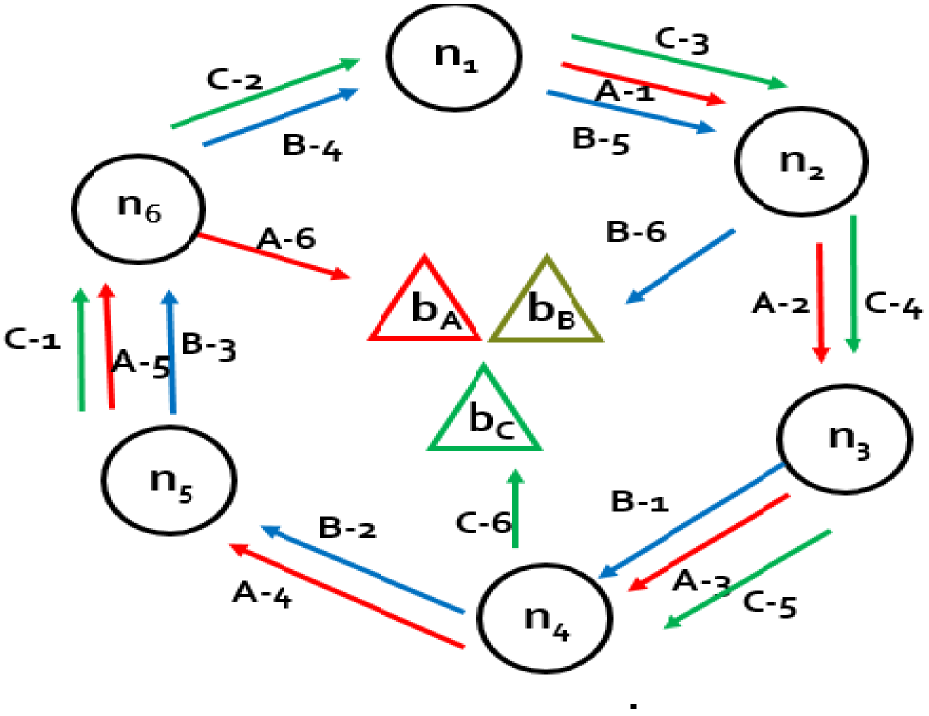

Figure2.2.1: Information accumulation in a system N with |N | = 9 hubs and k = 3 simultaneous information streams. Circles and triangles are speaking to IoT hubs and base stations, individually. Bolts are demonstrating the stream of information streams. The content by a bolt is demonstrating its information stream (i.e. A, B, C) and the schedule vacancy number (i.e. 1, •, 5).

2.2.3 Results and Discussions

The execution of the proposed organize structure is additionally assessed utilizing PC reproductions. In the reenactments, the span of an information accumulation process T with k simultaneous streams is utilized as the execution pointer. T is communicated as the aggregate number of availabilities required by the BS of various streams to gather information from every one of the hubs in the system. Reenactments were directed in Mat lab. In every reproduction, a system with |N | IoT hubs is considered.

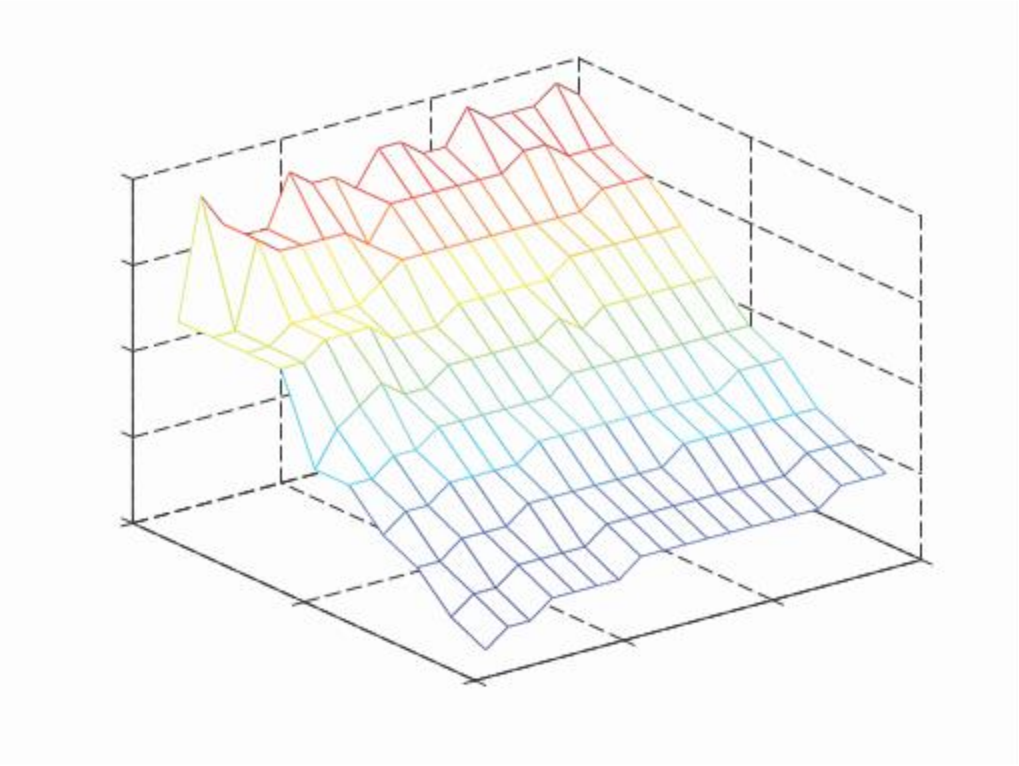

Figure 2.2.2: Information gathering terms of the proposed information accumulation tree in systems with |N | hubs and k information simultaneous information streams.

In the proposed organize structure, the procedure for acquiring its transmission timetables can be changed effortlessly to oblige other improvement requirements or criteria. One normal worry for versatile systems is the aggregate correspondence separation of the information accumulation tree, which may genuinely influence the lifetime of battery-controlled cell phones.

2.2.4 Conclusions

People in general and private web of things (IoT) frameworks are combined to shape an IoT organization. Under these interconnected frameworks, IoT gadgets will be shared among various gatherings. Various information gathering forms started by various clients can be completed on a similar arrangement of IoT gadgets all the while. In this paper, a deferral mindful system structure particularly intended for simultaneous information accumulation forms in IoT frameworks is proposed. The proposed arrange structure can abbreviate the deferrals of simultaneous information accumulation forms. Results in this paper demonstrate that the proposed thought can yield shorter information accumulation spans than a current information gathering system structure intended for a solitary information accumulation process.

2.3 Title: “Fault Tolerance in Wireless Sensor Networks” (2012)

A WSN is a self-composed system that comprises of an extensive number of minimal effort and low fueled sensor gadgets, called sensor hubs. It can be conveyed on the ground, noticeable all around, in vehicles, on bodies, submerged, and inside structures. Every sensor hub is furnished with a detecting unit, which is utilized to catch occasions of intrigue, and a remote handset, which is utilized to change the caught occasions back to the base station, called sink hub.

Sensor hubs team up with each other to perform undertakings of information detecting, information correspondence, and information preparing.

2.3.1 Disappointments in Wireless Sensor Networks:

a) Energy exhaustion

It have extremely restricted vitality and their batteries can't for the most part be revived or supplanted, because of unfriendly or perilous situations

b) Hardware disappointment

A sensor hub has two part: detecting unit and remote handset. Generally straightforwardly cooperate with the earth, which is liable to assortment of physical, substance, and organic variables.

c) Communication interface mistakes

Regardless of whether state of the equipment is great, the correspondence between sensor hubs is influenced by numerous variables, for example, flag quality, reception apparatus point, obstructions, and climate conditions

d) Malicious assault

It brings about low unwavering quality and execution of sensor hubs. Consequently, adaptation to non-critical failure is one of the basic issues in WSNs.

2.3.2 Fault Detection Techniques:

There are two kinds of fault location approaches: Self Diagnosis and Cooperative Diagnosis. Self-Diagnosis resembles Battery exhaustion and Cooperative Diagnosis resembles Failure in correspondence interface.

2.3.3 Fault Recovery Techniques:

WSN rebuilt or reconfigured, such that disappointments or defective hubs don't affect advance on arrange execution. The most usually utilized strategy for blame recuperation is replication or repetition of parts that are inclined to be disappointment. At the point when a few hubs neglect to give information, the base station still gets adequate information if repetitive sensor hubs are conveyed in the area. Hand-off Node Placement in Wireless Sensor Networks:

- Two-Tiered Wireless Sensor Networks
- Hop-by-Hop TCP for Sensor Networks
- Ride Sharing: Fault Tolerant Aggregation

2.3.4 Conclusion:

In the paper, compact examination is furnished with profitable information for future application to keep a similar sort of issues from happening. The lessons learnt from the distinctive arrangements can be utilized by any application. A characterization of the accessible adaptation to internal failure strategies for remote sensor systems has been proposed considering the different components received by the arrangements.

2.4 Title: “Distributed Construction of a Fault-Tolerant Network from a Tree”

The system structure amid the occasion of hub joins, hub flights or hub disappointment ought to keep up its own particular adaptation to internal failure and grants the reorganization of the tree. In this paper we talk about the difficulties by dispersed applications by introducing a calculation that effectively manufactures an intelligent blame tolerant system overlayed on a conveyed tree structure. Applications that don't require the "treeness" property can enable new hubs to join basically by connecting to one existing hub and run all correspondence on the blame tolerant overlay developed from the tree. For applications that require progressive system nature of trees, we show disseminated calculations that utilization the overlay to keep the fundamental tree associated within the sight of shortcomings. The calculation adjusts to hub joins, leaves and disappointments.

Build d-customary arbitrary charts where each hub has precisely d neighbors.

Such charts are hard to build and keep up in a dynamic conveyed setting. Consequently, we characterize another scope of irregular charts which we call (d,) - general arbitrary diagrams. These charts help in managing the dynamic idea of our system, while as yet accomplishing adaptation to non-critical failure. One result of utilizing this approach is that we don't expect hubs to tell different hubs in the system and when the hubs leave the system, along these lines pleasing disappointments.

At long last, we show a novel appropriated calculation that uses the overlay expander to keep the hidden tree associated within the sight of flaws. This calculation deals with most cases the calculation can effectively fix the tree when hubs bomb, be that as it may, in the impossible occasion of an expansive division of hubs flopping all the while or in some corner cases like the disappointment of the root hub, the calculation won't not succeed. In these cases we require a portion of the hubs to re-join the tree utilizing the default application-particular instrument. Expander diagrams are an all-around contemplated outline for blame tolerant systems.

2.4.1 Background material

In this area we exhibit some known outcomes from the hypothesis of arbitrary normal diagrams and irregular strolls.

a) Random customary diagrams

- Let S (n, d) signify the arrangement of all d-standard diagrams on n hubs and Gn,d be a chart inspected from S(n, d) consistently at arbitrary.
- Gn,d is an arbitrary consistent chart.
- It is realized that arbitrary consistent charts have asymptotically ideal extension with high likelihood.

b) Uniform testing utilizing arbitrary strolls

- An arbitrary stroll on a chart can be demonstrated as a Markov chain.
- For a chart containing n hubs, the likelihood change grid M of the arbitrary walk is a n × n matrix where every component Mij indicates the likelihood with which the irregular walk moves from hub I to hub j in one stage. Give πt a chance to be a vector to such an extent that $\pi t[i]$ is the likelihood with which the irregular walk visits vertex I at step t.
- Then $\pi t+1 = \pi tM = \pi 0M\ t+1$. A vector π is known as the station-ary circulation of the irregular walk if $\pi = \pi M$, i.e., the stationary dissemination continues as before after the arbitrary walk makes a stride, or any number of ventures so far as that is concerned.
- It is realized that an arbitrary stroll on an associated undirected chart with an odd cycle has an interesting stationary circulation.

2.4.2 Distributed expander development

We say a chart is (d,) - consistent if the degrees of all hubs in the diagram are in the range [d −, d]. At that point a (d,) - consistent irregular diagram, indicated Gn,d, , is a (d,)-

customary chart that contains a subgraph picked consistently at arbitrary from the set S(n, d−)— the arrangement of all (d−)- standard diagrams of n hubs.

2.4.2.1 Algorithm to generate (*d*,-)-regular random graph:

Each hub $x \in V$ executes the accompanying:

```
Introduction:
    $\Gamma G (x) \leftarrow \emptyset$
Primary:
    Repeat until the end of time
        If $|\Gamma G (x)| < d$
            Uniformly example hub y from V
            Send (Add: y) to x
After accepting (Add: y)
    If $y = x$ or $y \in \Gamma G (x)$
        Do nothing
    Else
        If $|\Gamma G (x)| = d$
            Pick z from $\Gamma G (x)$ at arbitrary
            Remove z from $\Gamma G (x)$
```

12. Send (Remove: x) to z

13. Add y to ΓG (x)

14. Send (Add: x) to y

After accepting (Remove: y):

15. Expel y from ΓG (x)

After accepting (Failed: y)

16. Expel y from ΓG (x)

2.4.3 Summary:

Our development of an expnder from a tree can be total marized as takes after:

- We develop (d,) - customary arbitrary diagrams from a tree. Every hub consistently tests hubs from the tree and adds them to its neighbor set, keeping up a most extreme of d neighbors.

- We utilize BIwalks to test hubs consistently at arbitrary from the tree. All BIwalks are begun from the root as this requires low message multifaceted nature for each refresh.

- As the expander develops, we can diminish stack on the root by utilizing MDwalks. MDwalks venture crosswise over edges of the expander. Our calculation brings about more MDwalks as the tree develops in estimate and turns out to be moderately steady.

Once built, the expander can be utilized by applications for blame tolerant correspondence notwithstanding when their straightforward joining technique brings about a tree. Alternate class of appropriated applications that utilize tree structures to abuse the inborn various leveled and non-cyclic properties, require the tree itself to be blame tolerant.

2.4.4 Tree remaking after disappointments

At the point when a hub z comes up short, the parent z of z just expels its fizzled youngster from ΓT (z) and sends the refreshed weight to its own particular parent (aside from when z is root), like the instance of a hub joining. No doubt a kid x of z likewise just needs to expel z from ΓT (x) and interface itself as an offspring of some haphazardly picked expander neighbor in ΓG(x).

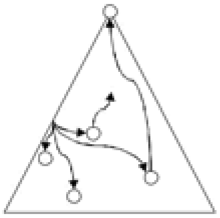

Figure 2.4.1: Tree support utilizing the expander. Triangle de-takes note of the tree. Little triangle signifies the subtree established at the fizzled hub z. Bended bolts indicate tokens sent by z's kid x to x's expander neighbors meant by dashed circles.

- The token (Tok: x, y) sent by x to its expander neighbor y is sent along the way from y to the root which at long last returns the token back to x.
- x sets y as its parent, unless the parent has just been set to another hub, e.g., on the grounds that the clock lapsed or in light of the fact that an alternate token was gotten from the root before.
- To maintain a strategic distance from complex situations that could bring about the arrangement of cycles, a hub x must dispose of tokens of the shape (Tok: x, y), if the token is sent to x by a youngster.
- In expansion, "nonces" ought to be utilized to recognize tokens sent crosswise over various keeps running of the convention. We preclude these points of interest from the pseudo-code for curtness.

3. SYSTEM DEVELOPMENT

The purpose was to model a fault recovery mechanism in the context of Concurrent Data Collection in IoT networks. The system is involved a conveyance of remote sensor hubs speaking with a focal hub serving to gather information from the system. The system experiences depletion of energy in hub groups nearest to the focal hub bringing about quickened loss of capacity, or system disappointment.

- The presumption is that in a thick sensor organize.
    - The occasion area will be shared by numerous sensors.

- The information among the sensors are gathered and contrasted with determine a choice.

- The defective sensor perusing will be covered by the lion's share of non-flawed sensors
    - There are numerous systems utilized as a part of written works

- Calculating the middle for a gathering of sensors

- Calculating the disappointment likelihood of a hub to settle on its future choice rightness.

3.1 Analysis for Fault Tolerance in Concurrent Data Collection Tree:

Case 1: Base Station becomes faulty.
Case 2: Node becomes faulty.
Case 3: Base Station and Node becomes faulty.

Some relations that will be needed are:

$$\tau_1 = \begin{cases} \lfloor \frac{2(|N| - u_{\max})}{(u_{\max}+1)} + 1 \rfloor, & \text{if } u_{\max} \text{ is odd,} \\ \lfloor \frac{2(|N| - u_{\max})}{u_{\max}} + 1 \rfloor, & \text{if } u_{\max} \text{ is even.} \end{cases}$$

$$\tau_2 = \begin{cases} \lfloor \log_2(|N| - \tau_1 \frac{u_{\max}+1}{2}) \rfloor + 1, & \text{if } |N| - \tau_1 \frac{u_{\max}+1}{2} > 0 \\ & \quad \text{and } u_{\max} \text{ is odd,} \\ \lfloor \log_2(|N| - \tau_1 \frac{u_{\max}}{2}) \rfloor + 1, & \text{if } |N| - \tau_1 \frac{u_{\max}}{2} > 0 \\ & \quad \text{and } u_{\max} \text{ is even,} \\ 0, & \text{otherwise.} \end{cases}$$

$$T = \tau_1 + \tau_2$$

3.1.1 Case 1: Base Station becomes faulty

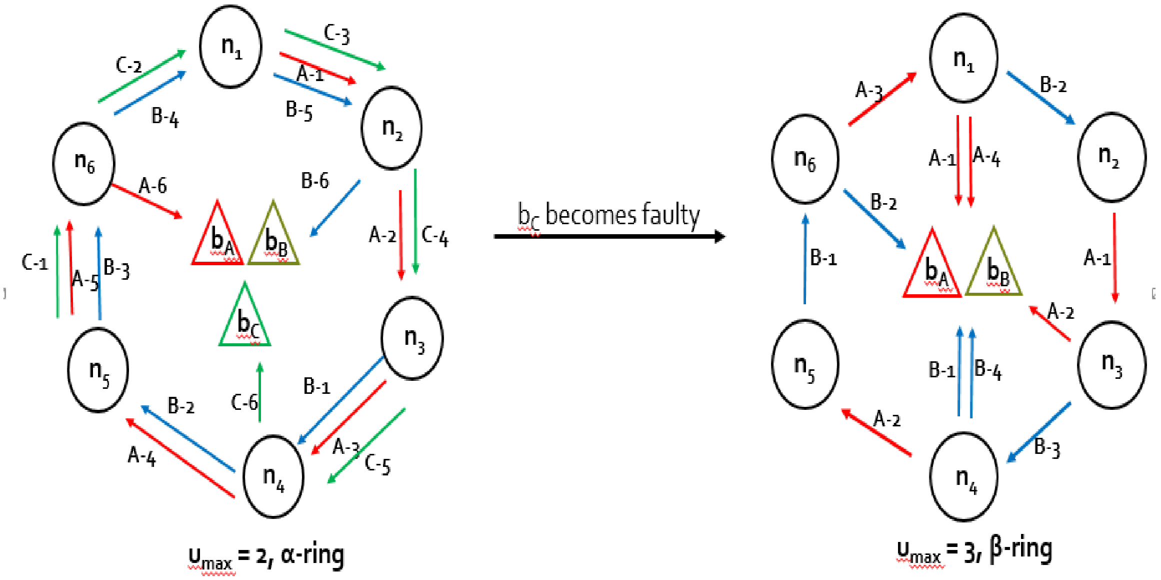

Figure 3.1.1 Illustrates the change in ring structure of concurrent data collection tree if a base station of the ring becomes faulty.

This will change the values of U_{max} and other values also:

$$\boldsymbol{u_{max}} = \left\lfloor \frac{|N|}{k} \right\rfloor,$$

$$c_1 = (1 + \mathrm{mod}(2(K-1) + t-1, |N_\alpha|))$$

$$c_2 = (1 + \mathrm{mod}(2(K-1) + t, |N_\alpha|))$$

For $K=2$, $t=4$, $T = 6$

$$c_1 = (1+\mathrm{mod}(2(2-1) + 4-1, 6)) = 6$$

$$c_2 = (1+\mathrm{mod}(2(2-1) + 4, 6)) = 1$$

$$c_3 = (1 + \mathrm{mod}(3(K-1) + 2(t-1), |N\beta|))$$

$$c_4 = (1 + \mathrm{mod}(3(K-1) + 2(t-1) + 1, |N\beta|))$$

$$c_5 = (1 + \mathrm{mod}(3(K-1) + 2(t-1) + 2, |N\beta|))$$

For Ƙ=2, t=3, T = 4

c_3 = (1 + mod(3(2-1) + 2(3-1), 6)) = 2

c_4 = (1 + mod(3(2-1) + 2(3-1) + 1, 6)) = 3

c_5 = (1 + mod(3(2-1) + 2(3-1) + 2, 6)) = 4

3.1.2 Case 2: Node becomes faulty

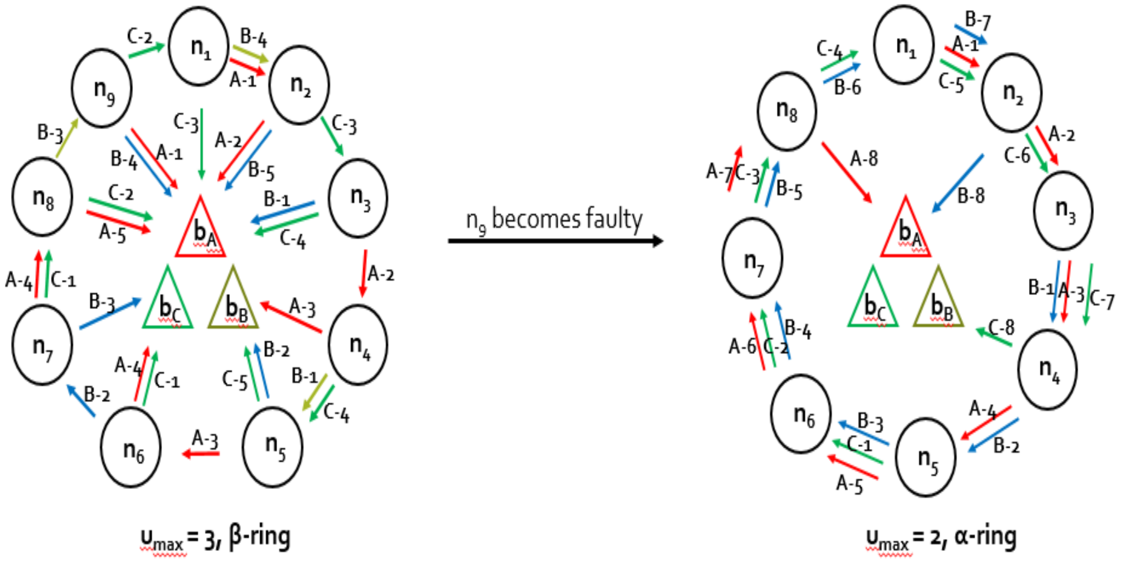

Figure 3.1.2 Illustrates the change in ring structure of concurrent data collection tree if a node of the ring becomes faulty.

This will change the values of c_1, c_2, c_3, c_4 and c_5:

For Ƙ=2, t=3, T = 5

c_3 = (1 + mod(3(2-1) + 2(3-1), 9)) = 8

c_4 = (1 + mod(3(2-1) + 2(3-1) + 1, 9)) = 9

c_5 = (1 + mod(3(2-1) + 2(3-1) + 2, 9)) = 1

For Ƙ=2, t=4, T = 8

c_1 = (1+mod(2(2-1) + 4-1, 6)) = 6

$c_2 = (1+\text{mod}(2(2-1) + 4, 6)) = 7$

3.1.3 Case 3: Base Station & Node becomes faulty

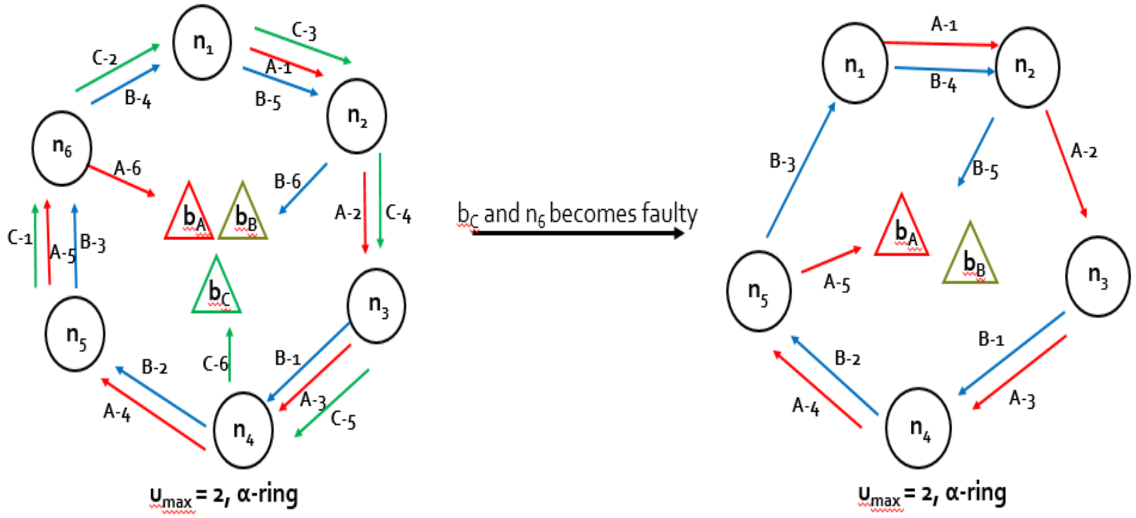

Figure 3.1.3 Illustrates the change in ring structure of concurrent data collection tree if base station and node of the ring becomes faulty.

This will change the values of c_1 and c_2 for different values of T:

For Ƙ=2, t=4, T = 6

$c_1 = (1+\text{mod}(2(2-1) + 4-1, 6)) = 6$

$c_2 = (1+\text{mod}(2(2-1) + 4, 6)) = 1$

For Ƙ=2, t=4, T = 5

$c_1 = (1+\text{mod}(2(2-1) + 4-1, 5)) = 1$

$c_2 = (1+\text{mod}(2(2-1) + 4, 5)) = 2$

3.2 Algorithm for Fault Tolerance in Concurrent Data Collection Trees:

```
For each hub x € V execute of the accompanying:

After getting (Failed: parent):

1.    parent ← ⊥
2.    start clock
3.    for every y ∈ ΓG (x)
4.        send (Tok : x, y) to y

After getting (TimerExpired : ):

5. Re-join the tree, set parent to new parent

After accepting (Tok : x , y ):

6.    if x is root
7.        send (Tok : x , y ) to x
8.    else if x = x and (Tok : x , y ) is sent by root

20

9.        if parent = ⊥
```

```
parent $\leftarrow$ y , refresh weight at y
stop clock
else
send (Tok : x , y ) to parent
```

4. PERFORMANCE ANALYSIS

4.1 About Contiki

Contiki is an open-source working framework that keeps running on the modest low-control microcontrollers which makes it conceivable to create applications that make productive utilization of the equipment while giving institutionalized low-control remote correspondence for a scope of equipment stages. We downloaded the Instant Contiki from contiki.org. So as to run instant Contiki we required VMware Player downloaded from vmware.com. Contiki OS is required as a result of three noteworthy reasons:

1. It backings a completely institutionalized ipv6 and ipv4 web norms and alongside this it additionally underpins the current low-control remote principles like 6lowpan, RPL and so on.
2. It backings the quick improvement. That is, with Contiki, advancement is simple and quick. Contiki applications are returned in standard C dialect and consequently it is anything but difficult to code.
3. It is an open source programming which can be uninhibitedly utilized as a part of business and non-business frameworks which can run numerous Internet of Things applications.

4.1.1 COOJA Simulation

COOJA is a network simulator designed for wireless sensor networks it is an open source simulator it is basically built in java and is capable of executing the C and C++ programs. COOJA has many advantages like:

- It is open source.
- It has simple UI flexible efficient and scalable.
- It uses basic C.
- It supports multiple platforms like sky mode etc.
- It has native TCP or IP support.
- It supports fully standardized IPv6 and IPv4.

- It supports low power standards like 6LOWPAN, RPL, CoAP etc.

4.2 Results

The above algorithm was implemented in Contiki Instant. It showed up different results for different failures.

4.2.1 Case A: Nodes becomes faulty

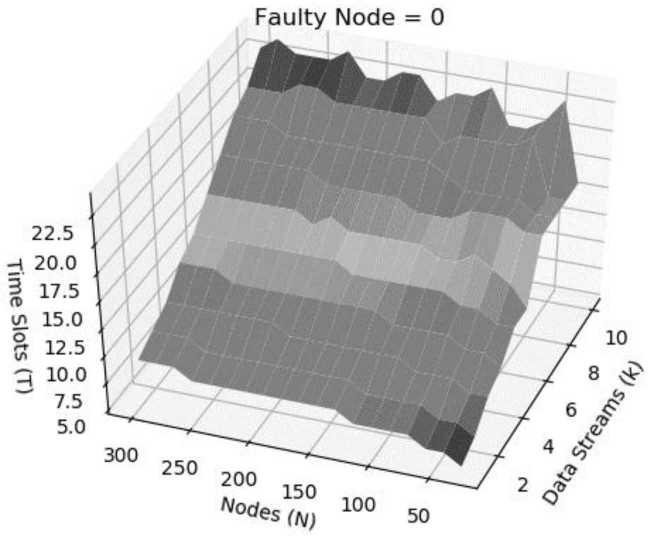

Figure 4.2.1 a) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if node 0 becomes faulty.

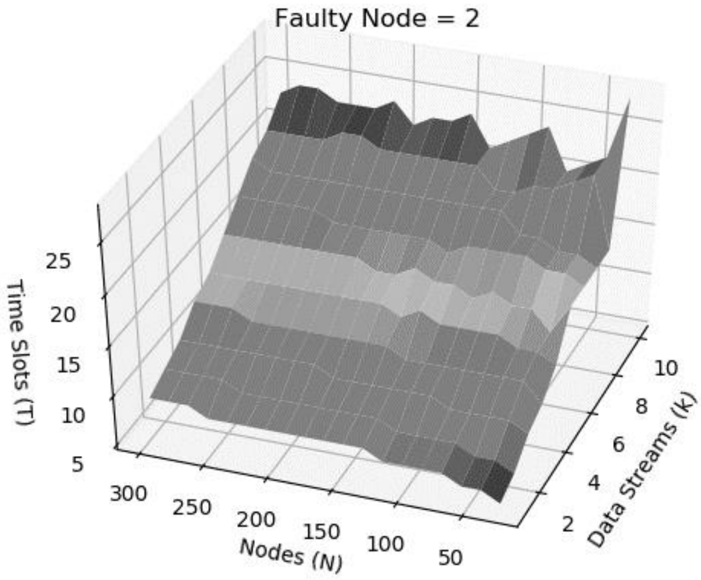

Figure 4.2.1 b) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if node 2 becomes faulty

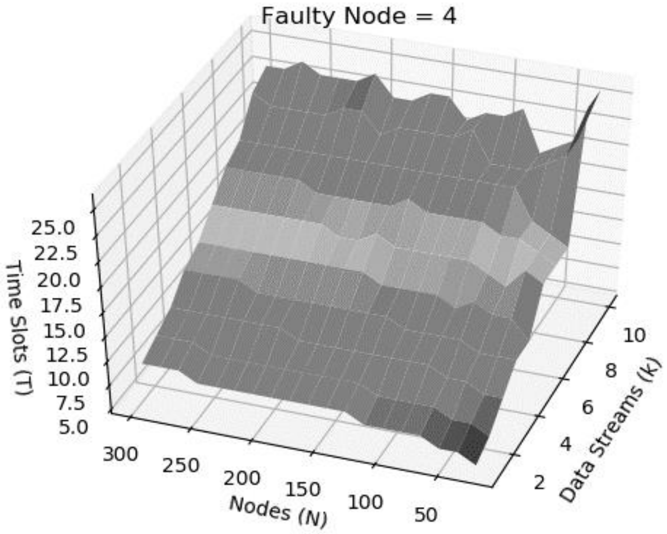

Figure 4.2.1 c) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if node 4 becomes faulty.

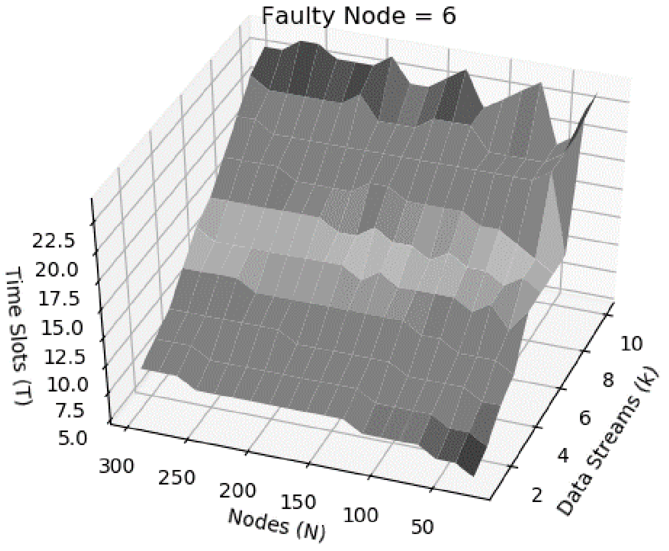

Figure 4.2.1 d) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if node 6 becomes faulty.

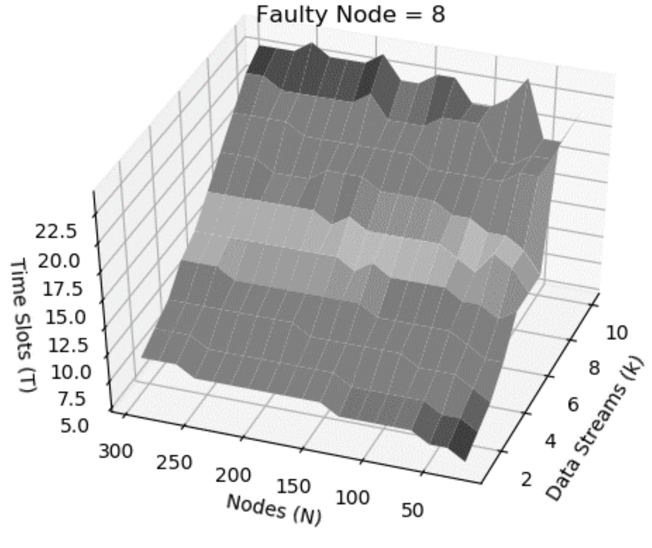

Figure 4.2.1 e) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if node 8 becomes faulty.

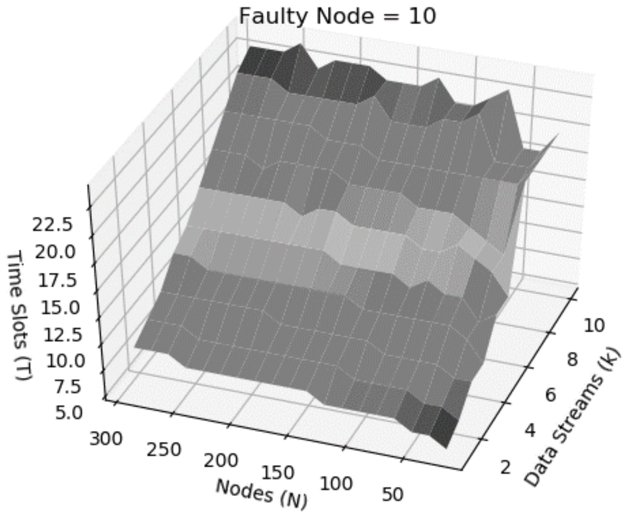

Figure 4.2.1 f) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if node 10 becomes faulty.

4.2.2 Case B: Base Stations becomes faulty

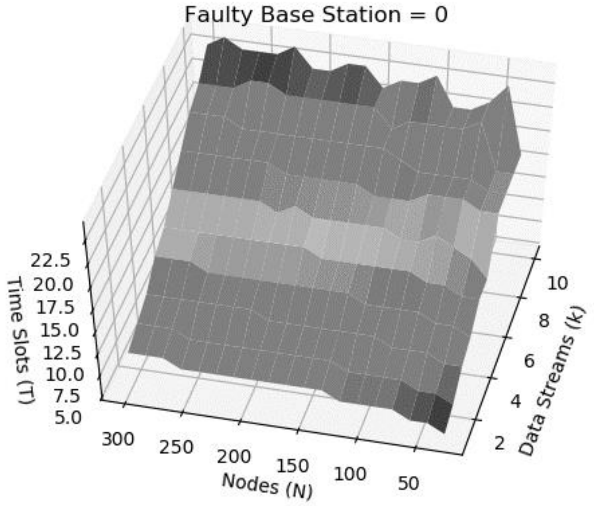

Figure 4.2.2 a) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 0 becomes faulty.

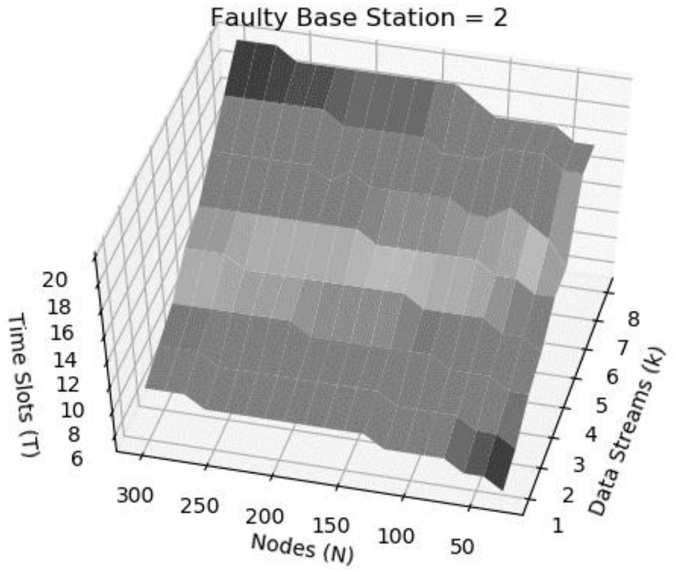

Figure 4.2.2 b) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 2 becomes faulty.

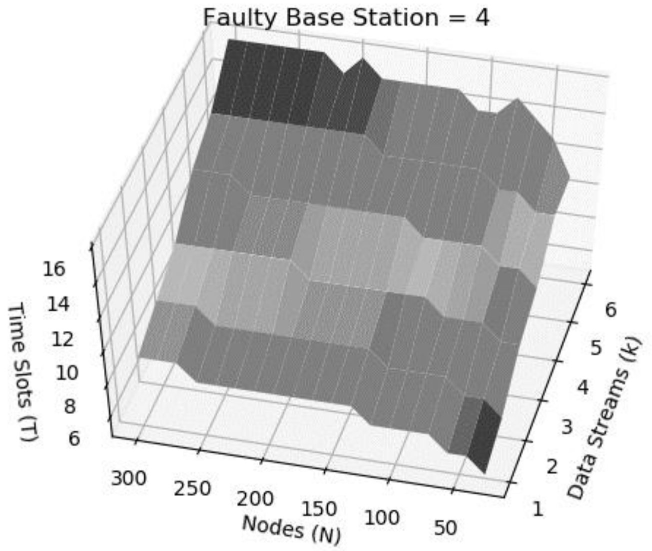

Figure 4.2.2 c) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 4 becomes faulty.

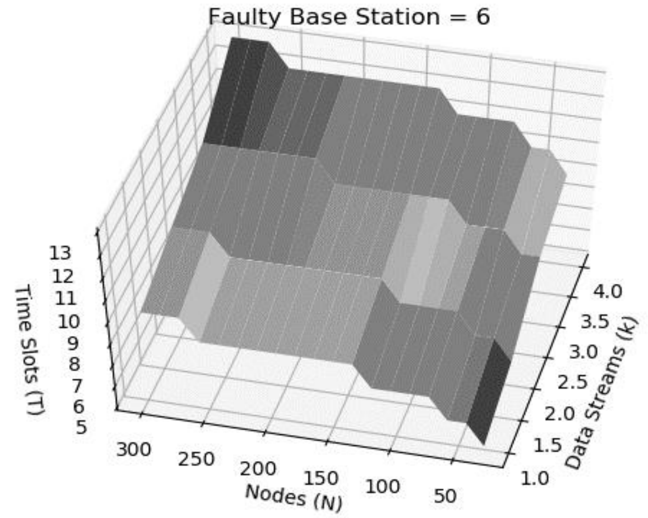

Figure 4.2.2 d) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 6 becomes faulty.

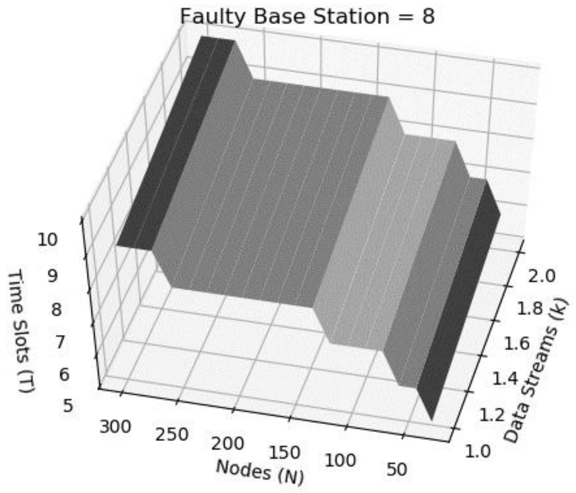

Figure 4.2.2 e) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 8 becomes faulty.

4.2.3 Case C: Nodes and Base Stations becomes faulty

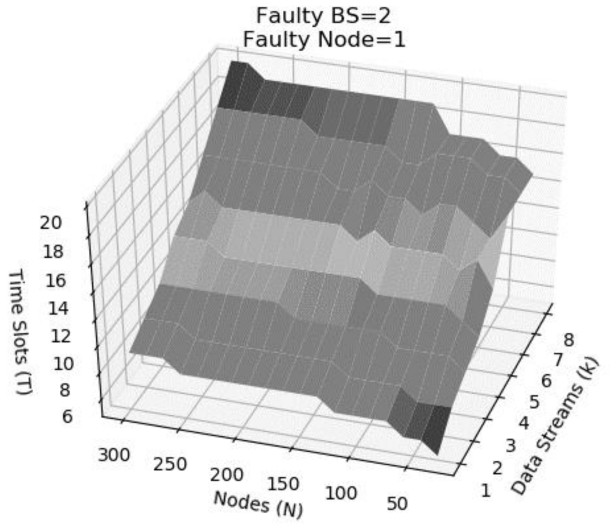

Figure 4.2.3 a) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 2 and node 1 becomes faulty.

.

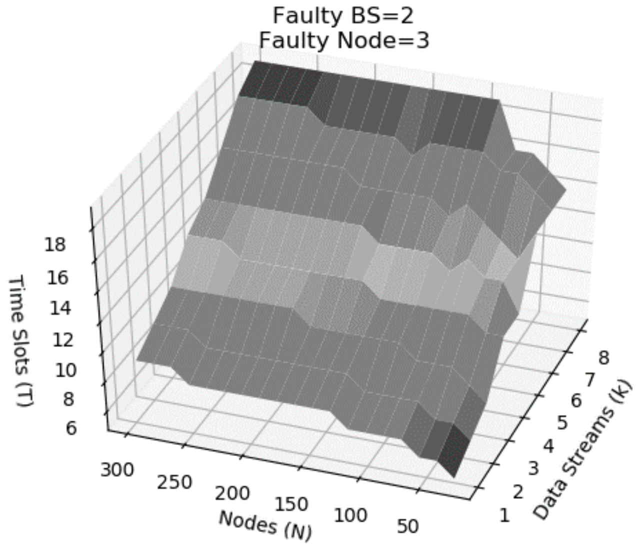

Figure 4.2.3 b) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 2 and node 3 becomes faulty.

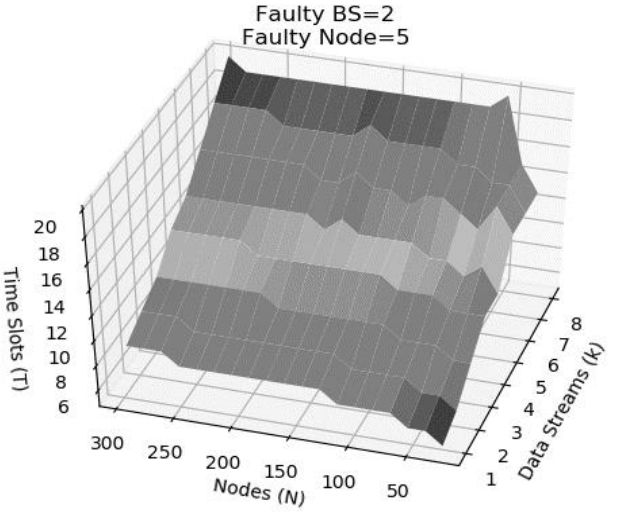

Figure 4.2.3 c) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 2 and node 5 becomes faulty.

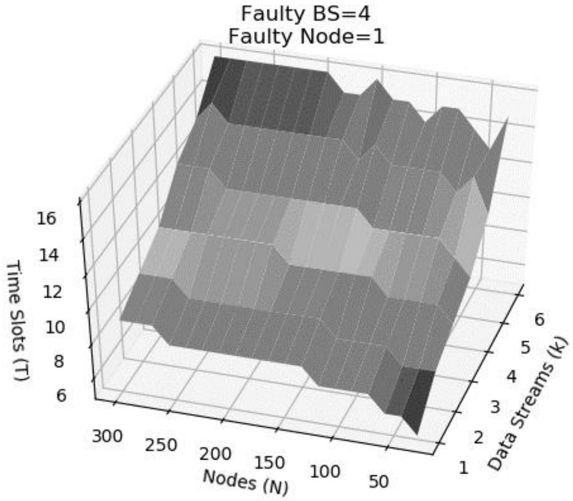

Figure 4.2.3 d) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 4 and node 1 becomes faulty.

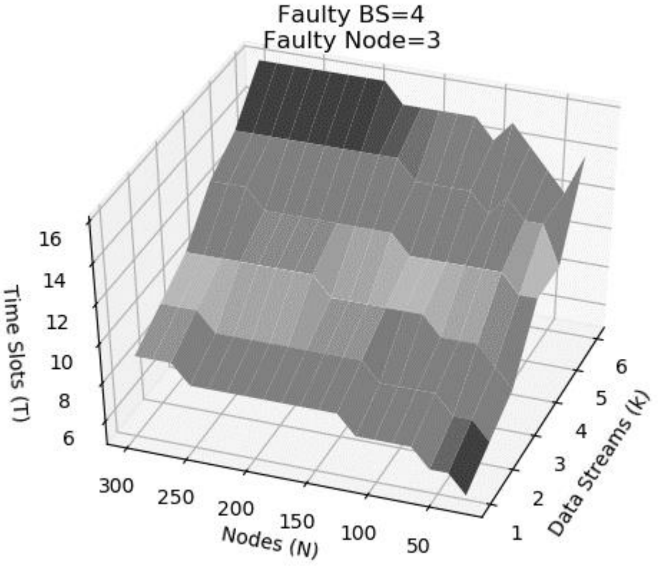

Figure 4.2.3 e) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 4 and node 3 becomes faulty.

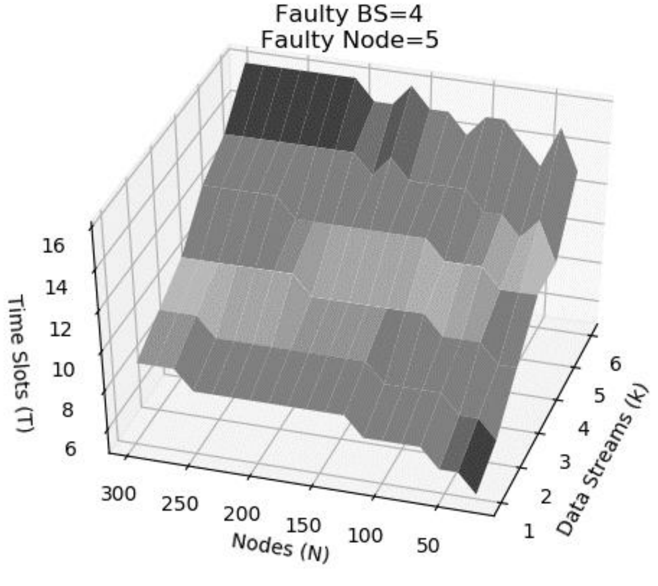

Figure 4.2.3 f) Graph showing analysis of Data Streams (k), Nodes (n) and Time Slots (T) if base station 4 and node 5 becomes faulty.

5. CONCLUSION

The sensor nodes may be easily damaged or depleted of energy altering the network topology and fragmenting routing paths. This dynamic characteristic of the network is especially critical to routing protocols where energy is lost in transmitting along failed routing paths. As noted above, sensor nodes are not readily replaced or recharged and hence the networks and employed protocols must complete their objectives in the presence of one or more failed nodes. This clearly establishes the value of employing mechanisms and protocols that persist correctly after the onset of network failures. This characteristic is referred to as fault-tolerance.

Fault prevention is the prior preclusion of the occurrence of faults in a system and is considered "almost impossible" to achieve Fault-tolerance accounts for the expectation of faults through the incorporation of redundancy to mitigate the faults and prevent system failure. Fault removal is the process of modifying the system to correct the fault and is considered maintenance and not germane to this discussion for the reasons noted above. Fault forecasting predicts the occurrence of faults and assesses the consequences but does not mitigate faults and is also beyond the scope of this work.

5.1 Conclusions

With ever increasing demand of IoT devices it becomes difficult to maintain a network with minimum delay and in real world fault free. Thus in this project we have successfully devised an algorithm for concurrent data collection trees that promise data collection time with minimum delay. This algorithm administers the problem of a faulty node as well as a faulty base station.

Appropriate terminology in the context of fault-tolerance have been defined and applied to an observed fault in the reference network model. Through simulation, the proposed recovery mechanism has been evaluated and demonstrated to improve the aggregate efficiency of the network during the transition from operational to a failed state.

4.1. Future scope

The future work involves work upon optimizing the fault tolerance algorithm for minimizing delays. With the developing enthusiasm for remote sensor systems (WSNs), limiting system delay and augmenting sensor (hub) lifetime are essential difficulties. Since the sensor battery is a standout amongst the most valuable assets in a WSN, productive usage of the vitality to draw out the system lifetime has been the focal point of a great part of the exploration on WSNs. Thus, numerous past research endeavors have attempted to accomplish tradeoffs as far as system postponement and vitality cost for such information total undertakings. As of late, obligation cycling method, i.e., intermittently turning ON and OFF correspondence and detecting capacities, has been considered to altogether decrease the dynamic time of sensor hubs and consequently broaden arrange lifetime. Be that as it may, this procedure causes challenges for information conglomeration. In this paper, we introduce a conveyed approach, named circulated defer effective information collection booking (DEDAS-D) to take care of the conglomeration planning issue in obligation cycled WSNs. The examination shows that our answer is a superior way to deal with tackle this issue. We lead broad reenactments to support our investigation and demonstrate that DEDAS-D outflanks other circulated conspires and accomplishes an asymptotic execution contrasted and brought together plan regarding information accumulation delay.

REFERENCES

[1] Chi-Tsun Cheng, Member, IEEE, Chi K. Tse, Fellow, IEEE, and Francis C. M. Lau, Senior Member, IEEE, “A Delay-Aware Data Collection Network Structure for Wireless Sensor Networks”(2011)- IEEE SENSORS JOURNAL, VOL. 11, NO. 3, MARCH 2011

[2] Chi-Tsun Cheng, Member, IEEE, NuwanGanganath, Student Member, IEEE, and Kai-Yin Fok, Student Member, IEEE, “Concurrent Data Collection Trees for IoT Applications”

[3] Sushruta Mishra, Dept. of CSE, GEC, BBSR, India, Lambodar Jena, Dept. of CSE, GEC, BBSR, India, AArti Pradhan, Dept. of CSE, GEC,BBSR, India, “Fault Tolerance in Wireless Sensor Networks”(2012) – International Journal of Advanced Research in Computer Science and Software Engineering. Available at: http://www.ijarcsse.com

[4] Michael K. Reiter, Asad Samar, Chenxi Wang, “Distributed Construction of a Fault-Tolerant Network from a Tree” - Proceedings of the 2005 24th IEEE Symposium on Reliable Distributed Systems (SRDS’05)

[5] http://www.contiki-os.org/start.html

[6] http://www.contiki-os.org/download.html

[7] https://github.com/contiki-os/contiki/wiki/Using-Cooja-Test-Scripts-to-Automate-Simulations

[8] https://github.com/contiki-ng/contiki-ng/wiki/Tutorial:-Running-Contiki%E2%80%90NG-in-Cooja

[9] https://gist.github.com/parhamr/6177160