DATA COLLECTION NETWORK STRUCTURE TREE FOR IoT

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Adarsh Kumar (191404)

Under the supervision of

Mr. Arvind Kumar

Department of Computer Science & Engineering and InformationTechnology
Jaypee University of Information Technology
Waknaghat,Solan-173234, Himachal Pradesh

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Data collection network structure tree for IoT"** 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 2022 to April 2023 under the supervision of **Mr. Arvind Kumar, Assistant Professor, CSE/IT**.

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream **Cloud Computing.**

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

Adarsh Kumar (191404)

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

Mr. Arvind Kumar

Assistant Professor

Computer Science & Engineering and Information Technology

Dated:

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ______________________ __**Department:** ______________ **Enrolment No** ________

Contact No. ________________________**E-mail.** ________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ____________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the project work successfully.

I am grateful and wish my profound indebtedness to Supervisor **Mr. Arvind Kumar,** Assistant Professor, Department of CSE & IT, Jaypee University of Information Technology, Waknaghat. The deep Knowledge & keen interest of my supervisor in the field of “**Cloud Computing”** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Mr. Arvind Kumar,** Department of CSE & IT, for her kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Student’s Name : Adarsh Kumar

Roll No. 191404

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviations	**Meaning**
IoT	Internet Of Things
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network
CDCT	Concurrent Data Collection Trees
MAC	Media Access Control
MC2H	Multiple Cluster 2 Hop
BS	Base Station
N2N	Node To Node Transmission
N2BS	Node To Base Station Transmission
TOCDCT	Time Optimal Concurrent Data Collection Tree
DADCNS	Delay Aware Data Collection Network Structure

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

Typically, an Internet of things (IoT) application consists of several smart devices that produce and exchange enormous volumes of data. Several applications can work together and share the same hardware infrastructure to satisfy their individual sensing requirements. Another advantage of such a shared network setup is that it benefits several subscribers to the same data. To boost productivity, the data that these devices produce is examined. Additionally, it is employed to enhance security and safety. A typical Internet of Things (IoT) network consists of several hundred interconnected devices, and numerous application processes rely on the data these devices produce. These applications collaborate and utilize the same device infrastructure to fulfill their individual sensing requirements in order to avoid over provisioning. However, this creates the issue of concurrent data collection. Multiple parallel data streams can be employed in concurrent data collecting methods to gather data effectively at numerous base stations. For IoT applications, the concurrent data gathering trees' present designs pose a number of additional difficulties. The delay optimization of the concurrent data collection procedures is one such difficulty. A Fast concurrent data collecting tree is suggested in this research. We demonstrate through simulations that the proposed design outperforms the current approach in terms of data collecting speed.

Chapter-1 INTRODUCTION

1.1 Introduction

70% of the world's population is projected to reside in cities by 2050. In order to support such rapid expansion, cities must use modern information and communications technologies to promptly provide up-to-date information to its residents. One of the technologies that is widely seen as a workable answer is the IoT. The bulk of existing smart cities feature independent, non-interoperable IoT systems. To maximize productivity and realise the full efficiency of smart cities, IoT applications installed on various location should be interconnected rather than several distinct closed form systems being built.

Additionally, IoT infrastructures in the public and private sectors should be shared to prevent overprovision and pointless redundancy. A smart city, for instance, often consists of numerous different applications, Hence different applications can interest in the information produced by these group of devices. Additionally, it's possible that different programmes may use the same data in different ways. Applications in such a situation stand to gain greatly from the shared infrastructure's reduced deployment costs, time, and maintenance requirements. Furthermore, using the infrastructure-as-a-service business model, similar infrastructure can also be made available to apps for a fee. Data collecting, privacy and data security, resource distribution, energy use, and a few other issues need to be addressed. Since numerous applications may query data sources at once and Data collecting offers a substantial challenge since the data's freshness must be maintained.

1.2 Problem Statement

The concurrent data collection trees are intended to minimise the latency when collecting data concurrently from the same set of nodes to multiple base stations. The network layout suggested in Delay Aware Data collection Network Structure Tree (DADCNS) [1] is demonstrated to be sub-optimal in terms of the quantity of time-slots required as well as the restrictions of the network to just the number of nodes in the value N = 2^p where p = 1, 2, 3, etc. Fewer time slots would be necessary for data gathering if more nodes could be used in a time-slot. To reduce the total time required for data collection, a delay aware time optimum concurrent data collection tree is presented in this project. Simulation results show that the proposed idea can drastically reduce latency in several data collection activities running at the same time.

1.3 Objectives

The main objective here is to optimise the time available for data collection. The more fresh the data is if it is supplied as quickly as it is collected. As a result, data must be gathered simultaneously at several stages of data extraction. The issue of concurrent data gathering in the IoT has only been addressed in a small number of recent research in this area.

Here, maximizing the amount of time spent collecting data is the main driving force. The more fresh the data is if it is delivered as quickly as it is collected. As a result, data must be gathered simultaneously at various points of data extraction. The issue of concurrent data collection in the IoT has only been addressed in a small number of recent works in this area.Two such algorithms that use various topological structures to deliver data concurrently are presented in [2] and [4] specifically.

In order to speed up the concurrent data collection processes, we further optimize the network structure in this paper. On the basis of the results of the simulation, the proposed structure for simultaneously data gathering can reduce the delay. One such approach that leverages several network structures to transmit data concurrently is presented in detail. In order to speed up the concurrent data gathering procedures, we further optimize the network design in this project. According to the results of the simulation, the proposed structure for simultaneously data collecting can reduce the delay.

1.4 Methodology

The purpose of the concurrent data collection trees is to reduce the amount of time that passes between collecting data simultaneously from the same set of nodes to different base stations. A tree structure has been proposed for the network structure. We can have any number of Nodes and Base stations and our model should provide a definite structure to ensure fast concurrent data collection from all the nodes. Nodes are sensors that collect information and data from the environments and have the computational power to store and forward that data to other nodes as well as base stations. Basestations are fixed transceivers that are the main communication point for one or more IoT devices. As multiple applications and processes may consume data sources simultaneously, the process of data collection presents a significant challenge. Concurrent data collection techniques are required because it is important to maintain the freshness of the data.

The topology we suggest is a tree structure made up of several clusters. These clusters are of the size 2^p where p=1,2,3…. For any number of inputs lets say N we divide the N nodes into different clusters of size 2^p. The clusters internally arrange themselves using the DADCNS algorithm.

The model will form a hierarchical structure where the cluster members will send data to their respective parent node. Similarly, the parent nodes will send the aggregated data from its child nodes to the cluster heads. The cluster heads will then aggregate the collected data and send it to the base stations.

1.5 Organization

The First chapter provides a brief introduction to the proposed problem and how we are planning to solve the problem. It also provides a brief overview of the current IoT domain and how fast data collection from sensors is extremely important in achieving zero delay real time information to various applications used in cloud or otherwise. The problem statement and the proposed methodology is briefly discussed in the chapter.

The Second chapter provides an overview of the various works on similar domain that was carried out previously by various authors. It was seen that various types of modes wereproposed using different algorithms and solutions to reduce the data collection speed of IoT devices as well as WSNs. The issue of concurrent data collection in the IoT has only been addressed in a small number of recent works in this area. One such algorithm that uses various network structures to deliver data concurrently is presented in [2] specifically. In order to speed up the concurrent data collection processes, we further optimize the network structure in this paper. According to the results of the simulation, the proposed structure for concurrent datacollection can reduce the delay.

An overview of the suggested model is provided in the third chapter. How the model was created mathematically and specifically. In this part, the recommended methodology is presented. WSN sensor devices are frequently used to meet a specific need or objective. On the other hand, with IoT networks, the devices may be shared by numerous parties or apps, which could result in data collection happening concurrently on the same set of devices.It becomes vital to consider concurrent data collection at several base stations as a result. Results from [2] showed that concurrent data collection trees (CDCT) had a lower data collection duration than the existing data collection network topology for a single data collection activity.

The fourth chapter provides the analysis of the algorithms of the proposed model using examples. Various simulations were performed on different formats of inputs. Firstly, the number of Nodes are varied and the number of base stations is kept constant. In the second case, the opposite is done. In both the scenarios the resultant timeslots are compared to CDCT and Time Optimal CDCT.

The Fifth and final chapter provides a summary of the entire project and what we can conclude from the simulations and observation. Fast data collection from IoT sensors can help receive data in lesser delay form ever before. which can eventually help benefit globally since IoT has become a day-to-day application in our lives. This project could be extended to various domains. For this particular research, we have only looked at the structure of network between the nodes and the base stations. In future we can also look into the frequency of each device and the energy consumption of these data collection processes and try to minimize it. This could in turn help us to reduce the use of energy to operate an IoT network. The future scope of the proposed project and its various applications are also provided in this chapter.

Chapter-2 Literature Survey

We have investigated quick information gathering determined to lessen the amassed information assortment plan length in [3, 9], and others have concentrated on it in [11, 14,16]. While the hypothetical parts were shrouded in [3], where we proposed consistent element and logarithmic estimate techniques on mathematical organizations, we exactly analyzed the impacts of transmission power guideline and different recurrence channels on the timetable length in [3]. (circle charts). A disseminated schedule opening task approach is recommended by Gandham et al. [15] to lessen the TDMA plan length for a solitary channel. Crude information convergecast has been analyzed in [15]. Moscibroda [16] researches the subject of joint planning and transmission power guideline for consistent and uniform traffic needs. Rather than the work referenced over, our own assesses transmission power guideline in genuine situations and processes lower limitations on the timetable length for tree networks utilizing calculations to arrive at these limits. We likewise think about the adequacy of different obstruction models and channel task methods, and we propose methodologies for building specific steering tree geographies that increment the information gathering rate for both collected and crude information convergecast. By designating schedule openings to hubs from the lower part of the tree to the top, symmetrical codes have been inspected for the purpose of diminishing obstruction. This guarantees that a parent hub doesn't send before it has gotten the bundles from its all posterity. This issue as well as the one that Chen et al. [17] alludes to combine projects of a single shot crude information. Since the quantity of bounces in a tree builds as its certification diminishes, the degree-obliged steering geographies we make in this study may not necessarily bring about plans with negligible dormancy. In this way, extra enhancement, for example, building limited degree, limited breadth trees, is required on the off chance that diminishing idleness is likewise a need. All skillet and

Tseng [17] furnish a concentrate thusly fully intent on limiting the most extreme idleness wherein they dispense every hub in a Zigbee network a signal period during which it can get information from its posterity. Tune et al. [13] presented a significant investment productive parcel booking method with intermittent planning for crude information convergecast. Nonetheless, they momentarily examine a 3-shading channel task instrument, and it is muddled whether the channels are frequencies, codes, or another strategy to take out obstruction. Whenever obstruction is wiped out, their calculation accomplishes the bound that we portray here. Furthermore, they make the suspicion that there will be no total obstruction from contemporaneous various shippers and that every hub will have a round transmission range. As opposed to their work, we consider various frequencies,assess the exhibition of three different channel task strategies, and survey the impacts of transmission power control utilizing practical obstruction and channel models, for example, an actual impedance model and covering channels, while likewise thinking about directing geographies. Tune et al. [13] developed their work and proposed a Macintosh convention in light of TDMA. To lessen blockage, TreeMAC considers the varieties in load at various levels of a steering tree and relegates time allotments in light of the profundity, or the quantity of bounces, of the hubs on the directing tree. This implies that hubs nearer to the sink are given a greater number of openings than their kids. Be that as it may, in light of the fact that the sink can get each three time allotments, TreeMAC works on a solitary channel and accomplishes 1/3 of the maximal throughput, like the limitations expressed by Gandham et al. [15]. Choi et al. [17] show that the test of limiting the timetable length for crude information convergecast on a solitary channel is NP-finished on broad charts. Lai's examination centers around expanding convergecast throughput by finding the briefest length, struggle free timetable. where the shippers are given time allotments by a ravenous chart shading method to keep away from obstruction. They likewise discussed how directing trees influence plan length and recommended a steering method called disjoint strips to send information along numerous briefest ways. Sending information across different pathways doesn't abbreviate the timetable, in any case, on the grounds that the sink keeps on being the bottleneck. As we will show in this work, using capacitated least traversing trees for crude information convergecast — where the quantity of hubs in a subtree is something like a portion of the all out number of hubs in the leftover subtrees — gives the steering structure a benefit. Both cell and impromptu organizations have gone through significant examination into the usage of different frequencies, Notwithstanding, there are a couple of studies that utilize a few directs in the field of WSN. To accomplish this, we survey the viability of three particular techniques that handle the channel task at different levels.

Energy protection becomes one of the primary issues in sensor networks because of the energy constraints of individual sensor hubs. In remote sensor organizations, remote correspondences represent a sizable measure of a hub's energy utilization. A transmission's energy utilization is contrarily corresponded with the correspondence distance. Thus, base station and hub significant distance correspondence is ordinarily deterred. Taking on a grouping strategy is one method for diminishing energy utilization in sensor networks [5]. A technique called bunching expects to bunch sensor hubs into groups. One hub is decided to act as the bunch head inside each group. The group chief is responsible for social occasion information from the bunch's individuals and intertwining it utilizing information/choice combination procedures. what's more, 3) detailing the melded information to a remote base station The sole hub in each bunch that participates in significant distance correspondence is the group head. Thus, the organization all in all utilizations less energy. There has been a ton of exploration done on the most proficient method to develop groups with the right organization engineering to save energy [6, [7, [8,] [9]. Filter is a grouping calculation that Heinzelman et al. contrived [6]. In Filter organizations, sensor hubs are organized in various bunch 2-bounce

(MC2H) organizations (premise station group head bunch individuals). The quantity of significant distance transmissions can be altogether diminished by utilizing the bunching idea. One more bunching calculation by the name of PEGASIS [7], set forth by Lindsey and Raghavendra, adopts a completely unique strategy by gathering sensor hubs into a solitary chain (SC) organization. In these organizations, just a single chain hub is picked as bunch head. How much energy utilized in significant distance transmission is additionally diminished by lessening the quantity of group heads. PEDAP [8], made by Tan and K. Orpeoglu, depends on the idea of an insignificant crossing tree (MST). The correspondence distances between sensor hubs are decreased, notwithstanding how much significant distance transmission. The assortment tree convention (CTP) was proposed by Fonseca et al. [9]. Anticipated transmissions (ETX) act as the directing slope in the CTP, a kind of slope based steering framework. The expected transmissions (ETX) of a parcel are the base expected for blunder free gathering [6]. High throughput is expected for ways with low ETX. In a CTP organization, hubs generally pick the course with the most minimal ETX. The ETX of a way is normally conversely connected with the related way length [18]. Accordingly, CTP can essentially abbreviate the correspondence holes between sensor hubs. These calculations have potential energy-saving advantages. An organization made by an energy-proficient grouping procedure, in any case, wouldn't generally be liked for information gathering. In Segment V, a presentation examination of these organization structures' information gathering adequacy will be introduced. The objective of this examination is to look at the adequacy of information gathering in networks made by different bunching strategies. Occasion setting off calculations like Adolescent and APTEEN will not be considered in this exploration. The connected concentrate on powerful information gathering was finished by Florens et al. [10].

Lower limitations on information gathering time for various organization structures are determined in their work. The effect of information combination, which is believed to be one of the critical attributes of sensor organizations, was not considered. For remote sensor organizations, Wang et al[11] .'s recommended connect booking calculations that can altogether increment network throughput. In any case, it is accepted in their work that information linkages between remote sensor hubs are predefined. Conversely, the objective of this paper is to make information linkages between remote sensor hubs to decrease the time utilized for information gathering. Solis and Obraczka finished one more similar piece of work in which they researched the impacts of timing on information total for sensor organizations. Chen et al. [17] investigated the impacts of organization limit under various organization limit under various organization structure and steering calculations. As indicated by their examination, limit alludes to how much start to finish traffic that an organization can uphold. Their principal stress isn't a defer in the information assortment method.

Early work of Cheng et al. [17] analyzed the issue of information gathering in huge scope tangible frameworks. In their exploration, they considered assembling information from a sizable number of individuals to a solitary information extraction point. The creators of [17] have offered a new viewpoint on the very much contemplated directing issue in PC networks in light of the perception that every one of the hubs in a tangible framework are possessed by a similar party. That is, one ought to enhance the utilization of his organization assets by laying out a planned transmission plan as opposed to keeping away from swarmed channels. A structure for evaluating the time execution of information assortment and information dispersion undertakings in tactile frameworks was presented by Florens et al. in [10]. At the point when they work, For networks with various geographies, they found low limits and given the matching ideal transmission plan. The primary specialists to zero in on the issue of nonstop information assortment in tangible frameworks are Ji et al. They fostered the lower limits for single-depiction information assortments and persistent information assortments in their work. They likewise exhibited how the utilization of gadgets with various handsets may significantly accelerate an information assortment activity. The previously mentioned endeavors filled in as the reason for [10], [11]'s production of postponement mindful information gathering network geographies. A postponement limited network geography for fusible information and its going with development strategies for concentrated and conveyed frameworks were presented by Cheng et al. in [17]. One more organization geography was presented by Cheng et al. in [17] to empower deft in-network information combination. which a star organization's top bound can never outperform. A deferral mindful organization geography and its development calculation have been introduced in to defeat clashes between transmission plans for tactile frameworks that request consecutive information gathering tasks. Chen et al examination of the information gathering issue in [17] considered channel models. They laid out upper and lower constraints for information assortment procedures in networks where information combination isn't proper in their work. A speedy information conglomeration tree for single-depiction information gathering in remote sensor networks was advanced by Durmaz Incel et al. in [3]. Obstructions between sensor hubs are viewed as all through the tree development process. Wang et alstrategy .'is involved getting a By picking just a portion of the hubs for testing, roughly acquiring information. For cases where information show a serious level of spatial connection, their proposed arrangement is very viable and trustworthy. Enhancing transmission plans for tactile frameworks with dynamic traffic designs has as of late been thought about by works in [19]. In [20] and [21], probabilistic organization models for tactile frameworks were contemplated. [18] has explored the information assortment techniques in tangible frameworks with portability. In any case, the work booking issue in remote sensor organizations (WSNs) was analyzed in progress by Kapoor et al. which might have the most similitudes to the issue viable in this work. It is underscored that the really empowering innovation that recognizes the Web of Things (IoT) from conventional tactile frameworks is machine-to-machine (M2M) correspondence. Sensor hubs in common WSNs are regularly possessed and worked by a solitary element. Be that as it may, in IoT applications, IoT gadgets may be shared by various clients or projects, who can then begin various simultaneous information conglomerations on similar arrangement of hubs. Sadly, none of the works referenced above consider various information extraction destinations and simultaneous information assortment tasks. Late years have seen a ton of interest in information assortment in genuine IoT frameworks that a worldwide scaled IoT league might be accomplished by utilizing satellite information lines to connect separated unique IoT pieces. Albeit a typical parcel misfortune pace of around 25% is expected in IoT frameworks, delays welcomed on by retransmissions can be limited by compacting information and dispensing with bundle fragmentations.Wu et al. guaranteed that when gadgets are sharing a solitary channel, information gathering frameworks in light of the ordinary IEEE 802.11 standard might encounter execution debasements. For viable information gathering in huge IoT frameworks, they introduced a versatile channel portion strategy and an energy-mindful access control convention.

By intertwining versatile specially appointed networks (MANETs) with WSNs, Bellavista et al. are the initial in the field to bring versatility into IoT [22]. The essential objective of the information assortment calculations made for remote sensor organizations (WSNs) was to broaden the organization lifetime by forfeiting information assortment efficiency.In a few different ways, a directing convention is likened to an information assortment strategy since it tries to make a course that hubs can follow to pass information on to an extraction point.In [5], a rundown of these steering calculations is shown. As an alternative,[6]-[7] concentrates on focused on decreasing energy use by laying out bunches. For information gathering, different tree geographies including chain, least spreading over, and group trees were thought about. On different geographies and organization structures, Florens et al. [10] assessed a lower destined for the information gathering time in the wake of breaking down the viability of information gathering. The impact of time on information conglomeration for sensor networks is examined in [9] and [10]. By diminishing correspondence costs, Yin et al. presented an information total tree for modern inquiries. To boost the total addition, they considered both the information pruning power and the collection cost in their work. An information gathering design in the Web of Portable Things as a help stage was tended to by Maiti et al. in [23]. They talked about how information gathering connects with the amount of sensors, energy the board, information security, and information quality in this work. For multi-jump, gadget to-gadget (D2D) correspondence with translate and advance transferring, Chen et al. [17] laid out an ideal directing in view of reliable network likelihood by thinking about fixed and arbitrary areas of base stations. Utilizing an alliance development game, Xu et al. made a multi-jump directing system to limit network delay. It is exhibited that the game offers a Nash-stable harmony. In [17], Chen et al. utilized a Poisson circulation to examine the exhibition of multi-jump directing. At a solitary base-station, Kolcun et al. introduced a circulated information gathering component in [24]. To give speedier information gathering by choosing substitute directs for information move in case of organization clog, Cheng et al. [17] presented an organized information assortment framework. Ji et al[20] .'s constant information assortment technique depended on utilizing all suitable organization assets. An organization format that separates the hubs into bunches of changing sizes and permits the hubs in each group to on the other hand communicate to the base station is proposed by Cheng et al. in [17]. To accelerate the information assortment process, they have additionally improved the between group correspondence distances.

CONCURRENT DATA COLLECTION TREES

Consider an Internet of Things (IoT) network N = n1, n2, , n|N| and a collection of base stationsS = s1, s2, , s|S|. It is expected that each of these |N| IoT nodes can connect to the base stations and communicate with one another. Data gathered from diverse IoT devices is thought to be perfectly fusible since numerous incoming data packets can be combined into one before being delivered to one's parent node [6]. It is anticipated that the time required to combine the data will be minimal because a single unit of data will be transmitted throughout a single time slot. There are k total concurrent data streams, and each concurrent data aggregation process will use a different base station (BS) to connect to the IoT network.

The issue of simultaneous information gathering utilizing a few information streams at various base stations is first tended to by the creators in Quite a while Assortment Trees. The previously mentioned strategy depends on the plan of CDCT, which is portrayed as rings, otherwise called α-rings and β-rings. The information gathering time from similar arrangement of hubs to many base stations is abbreviated because of this organization structure. The organization structure is predicated on the possibility that hubs can join information from numerous IoT hubs into a solitary bundle prior to sending it on. It is guessed that a solitary piece of information will be passed on once. Such information collection tasks can all work all the while to an alternate IoT base station. The aggregate sum of base stations is identical to the aggregate sum of dynamic information gathering tasks, which is shown by the letter k. Also, it is accepted that the organization's transmissions are adjusted. All in all, various transmissions can happen at the same time between non-covering sets of hubs. The proposed network setup utilizes the absolute most hubs during each schedule opening, contingent upon the amount of hubs and information streams. Moreover, every information stream inside a time allotment should involve a special blend of hubs for transmission. Umax determines the greatest number of hubs that can be utilized by an information stream in its most memorable schedule opening.

All simultaneous information streams ought to begin and quit during a similar time span to ensure reasonableness among these clients. In any case, each schedule opening in equal information streams ought to utilize similar number of hubs. Every information stream ought to utilize the best number of hubs at each schedule opening to speed up the general information assortment process.

TIME OPTIMAL CONCURRENT DATA COLLECTION TREES

In contrast to Wireless Sensor Networks (WSNs), the Internet of Things (IoT) allows many applications to share the same device infrastructure. The devices can be queried by multiple such apps at once, which might necessitate starting concurrent data streams on the devices. The authors have noted this problem in [2]. To conquer this, they introduced a simultaneous information gathering tree structure known as α-rings and β-rings that is addressed as rings. These rings are utilized to make the organization engineering, and information is gathered all the while at a few base stations (BSs) utilizing similar arrangement of hubs. Subsequently, it's urgent to capitalize on the hubs all through a specific time span. To do this, [4] focused on the β-rings as opposed to the α-rings to amplify hub usage.

The Time Optimal CDCT network structure that limits the amount of schedule openings required for simultaneous information gathering is characterized here. The organization setup comprises of various gadgets or hubs, meant by $N = \{n1, n2, . . . , n|N|\}$ . . base-stations (BS) $S = \{s1,$

s2, . . . , s|S|} are utilized to represent, n|N| that are shared by numerous applications. s1,s2,s3 . . , s|S|. We consider a solitary jump network design in which gadgets can associate straightforwardly to the BS. A gadget can likewise converse with some other gadget in its area. Accepting that the information delivered by these gadgets is connected, gadgets can join and impart information. Simultaneous information assortment is important on the grounds that various applications might require information without a moment's delay.. All gadgets in N are expected to have an information to send, and are engaged with information transmission. It is accepted that each gadget in the organization N is taken part in information transmission and has an information to send. A specific number of simultaneous datastreams k are begun in the organization relying upon the number of these equal applications are mentioning information. Each schedule opening is normally considered sending one unit of information. The proposed network geography utilizes the most gadgets in the first $\tau 1$ time allotments, contingent upon the amount of gadgets and information streams. The gadgets utilized for information transmission of different information streams change contingent upon the time allotment. Umax determines the most gadgets that can be utilized by an information stream in the initial time slot.

Chapter-3 SYSTEM DEVELOPMENT

The proposed network structure is a tree structure. We can have any number of Nodes and Base stations and our model should provide a definite structure to ensure fast concurrent data collection from all the nodes. Nodes are sensors that collect information and data from the environments and have the computational power to store and forward that data to other nodes as well as base stations. Base stations are fixed transceivers that are the main communication point for one or more IoT devices. The process of data gathering is a significant difficulty since various applications and processes may simultaneously consume data sources, and because data freshness must be maintained, concurrent data collection approaches are required.

3.1 INTRODUCTION

We propose the network architecture that minimises the quantity of time slots needed for simultaneous data collection.The network architecture consists of a set of devices/nodes, represented by $N = \{n1, n2, . . . , n|N|\}$ that are shared among a large number of different applications, represented by set of base-stations (BS) $K = \{k1, k2, . . . , k|K|\}$. We consider a multihop network structure, where devices can reach the BS through their cluster head(CH) only. Each device can only communicate with a parent device only. Devices can aggregate and transmit data because it is assumed that the data these devices produce is correlated.

Concurrent data collection is necessary because multiple applications may need data at once. It is assumed that every device in the network N is engaged in data transmission and has some data to send. Initiating such a number of simultaneously data streams in the network depends on how many of these parallel applications are requesting data. Each time-slot is typically thought of as transmitting one unit of data. The suggested network topology makes use of the most devices in T time-slots, according to the quantity of devices and data streams. The devices used for data transmission of various data streams vary depending on the time slot.

Data gathered from diverse IoT devices is presumed to be completely fusible since numerous received data packets can be fused into one before being delivered to one's parent node. It is anticipated that the time required to combine the data will be brief because just one unit of data will be transferred during one time slot. To preserve fairness among these users, all concurrent data streams should begin and stop at the same time. However, parallel data streams should employ the same number of nodes for each time period. For the best use of the overall data collection process, each data stream should utilize the highest number of nodes at each time-slot.

3.2 DELAY AWARE DATA COLLECTION NETWORK STRUCTURETREE (DADCNS)

Before going into our model development, we need to understand DADCNS. This is because we used DADCNS to form the various clusters in our model. We have overcome a specific weakness of this model. This will be later discussed.

A multiple hop tree structure makes up DADCNS. For the highest data collecting efficiency, the network structure's node count N must be kept to a maximum of $N = 2^p$, where p = 1, 2,... Each member of the cluster will be given a rank, which is an integer with a value between 1 and p. When connecting to nodes of rank k-1, which have ranks ranging from 1, 2,..., k-1, a node of rank k will build k-1 data linkages. All of these k- 1 nodes will be offspring of the node with rank k. A node with rank k and a node with a higher rank will be connected via a data link. This node of higher rank will be the parent node of the node with rank k. We will treat the cluster head as an exception. The cluster head is the network node with the highest rank. Instead of connecting to a node of higher rank, the cluster head will establish a data link with the base station. The rank distribution will follow an inverted exponential base-2 function using this logic, as seen in Table 3.1.

TABLE 3.1

CLUSTER MEMBERS' RANK DISTRIBUTION IN DADCNS STRUCTURE WITH NETWORK SIZE N = 2^p,WHERE p = 1, 2, · · · .

Rank	1	2	$\cdots$	$\log_2 N - 1$	$\log_2 N$
No. of nodes	$\frac{N}{2^1}$	$\frac{N}{2^2}$	$\cdots$	$\frac{N}{2^{(\log_2 N-1)}}$	$\frac{N}{2^{(\log_2 N)}}$

Fig 3.1 displays an example of the DADCNS network with a size of 16. In this illustration, thebase station needs 5 Time Slots to gather all of the data from the 16 nodes.

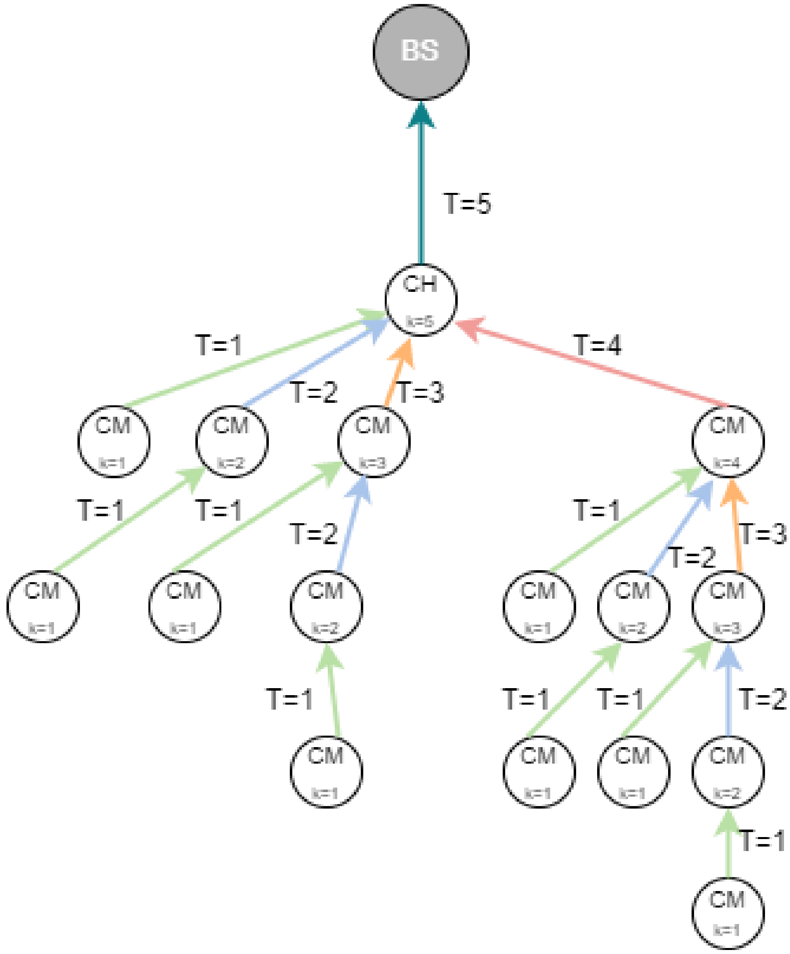


Fig. 3.1. This represents the structure inside one cluster of network size N 16.The nodes of the cluster are arranged using DADCNS. The cluster's members are shown as circles with CM. The cluster head is represented by a circle with CH. The base station is represented by a filled circle with BS. The variable k represents the rank of each node. The existence of a data link is represented by a dashed arrow, and the direction of the arrow indicates the direction of data flow.

By dividing the time quantum into time slots of durations T, the above process will take 5 timeslots.

3.3 LIMITATIONS OF DADCNS

The main limitation of DADCNS is that this network architecture can work only against a number of nodes in the form of N= 2^p where p =1,2,3…. This makes the mentioned algorithm very restrictive. In the real world the number of nodes will most likely be not in the form of 2^p. So the proposed algorithm in the project is made to remove those limitations as well as perform efficiently in gathering data from any number of nodes and forward it to any number of base stations. The proposed model is explained below.

3.4 PROPOSED DATA COLLECTION TREE STRUCTURE

Our proposed structure is a tree structure of a number of clusters. These clusters are of the size 2^p where p=1,2,3…. For any number of inputs lets say N we divide the N nodes into different clusters of size 2^p. The clusters internally arrange themselves using the DADCNS algorithm. The model will form a hierarchical structure where the cluster members will send data to their respective parent nodes. Similarly, the parent nodes will send the aggregated data from its child nodes to the cluster heads. The cluster heads will then aggregate the collected data and send it to the base stations.

The base stations are numbered serially based on a number of factors. The cluster heads will access the base station serially.

Take a look at a cluster where N = 2^p, where p = 1, 2,...,... The number of time slots needed fora node of rank k = 2 to gather data from all of its child nodes is equal to the number of child nodes it has, which is 1. The case for $k = 2$ is thus supported. Let's now assume that each node in a connection with $k = n$ needs $n-1$ time slots to gather all the data from its child nodes. Node I has n directly connected child nodes, and its rank is $k = n + 1$. The ranks of each of these directly connected child nodes range from 1 to n.

Since data packets produced by sensor nodes are considered to be highly correlated, a node is always capable of combining all packets it receives using data/decision fusion techniques into a single packet.

Using DADCNS, for each cluster with G nodes with $G= 2^p$ where $p=1,2,3\ldots,$ the cluster head is the only node with the highest ranking which is

$$kmax = \log2(G) + 1 \qquad (1)$$

Similarly, the number of time slots t(G) required for a cluster head, with rank kmax, to collect data from all its child nodes is

$$t(G) = kmax - 1 = \log2(G) \qquad (2)$$

Therefore, the total number of time slots required by one cluster head to send data to the 1st base station is,

$$t(G) = \log2(G) + 1 \ (3)$$

By adopting the proposed network structure, the number of time slots T(N) required for Knumber of base stations to collect data from the whole network with N nodes is given by

$$T(N) = floor(\log2(N)) + K (4)$$

Or

$$T(N)=M+K \qquad (5)$$

Where M is the number of clusters.

Figure 3.2 shows a network with N devices which has been divided into clusters of 16 nodes, 8 nodes, 4 nodes and 1 node respectively. All these 4 clusters follow DADCNS algorithm internally with ranking each node and its subsequent child nodes.

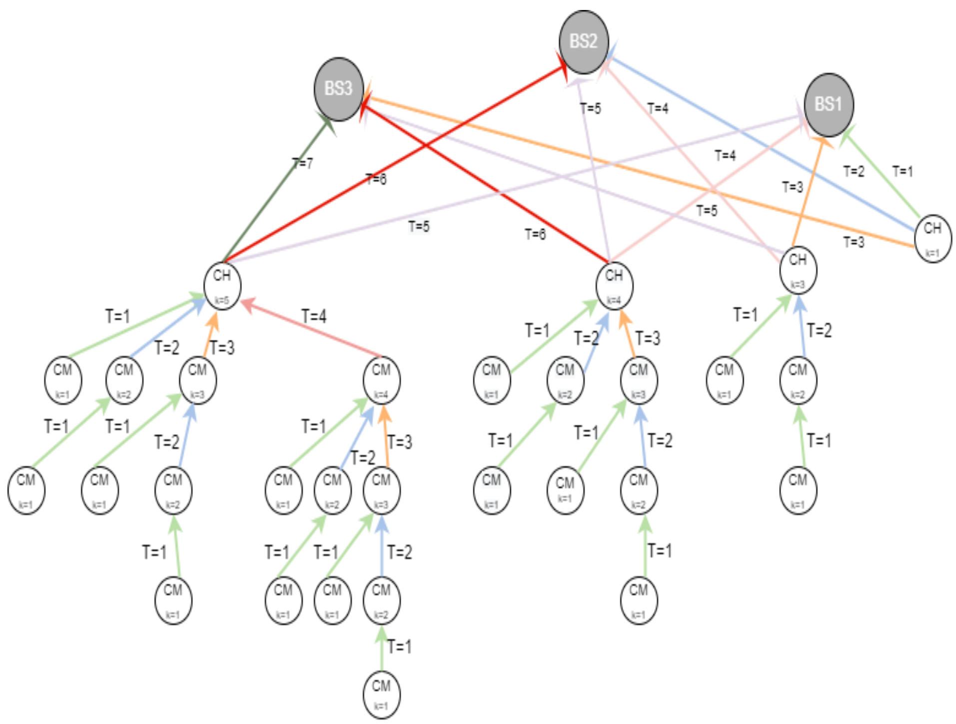


Fig. 3.2. Network size N = 29 for the proposed network structure. The cluster's members are shown as circles with CM. The cluster head is represented by a circle with CH. The base station is represented by a filled circle with BS1,BS2,BS3 . The variable k represents the rank of each node. The existence of a data link is represented by a dashed arrow, and the direction of the arrow indicates the direction of data flow.

As we can see, the total time slots taken by this proposed structure to send all the data from all the nodes of all the clusters to all the base stations in 7 time slots. Parallel data communication is maintained. The structure is also a multi hop structure and the number of hops vary from cluster to cluster depending on its size and kmax values. The next section will prove the algorithms for the proposed design.

Finally, we have to manage how each cluster head will access the base stations. To avoid collisions the proposed structure will arrange the base stations serially such that all the cluster heads will access the base stations in the same order. Which means, by the end of the operations all the base stations will be accessed a similar number of times which will be equal to the number clusters formed in the proposed model. Moreover, in our model all the clusters will be of different sizes so no two cluster heads will finish collecting data from all its nodes at the same time. This means that no two or more clusters will access one particular base station at the same time slot if all base stations are accessed serially in the same order by all the cluster heads. This ensures that there would be no collisions in communication between cluster heads and base stations.

3.5 ALGORITHMS

Let us consider a network of N devices and number of Base stations K. To divide and allocate all the nodes/devices into clusters we use Algorithm 1. The number N is subtracted by the 2^floor(log2(N)) and the resultant number is subtracted from N to find the next cluster size. This process is continued till N becomes zero which means all the nodes are allocated into clusters.

Algorithm 1: (Division of all nodes into clusters)

1. Create an array A to store the Different clusters of each size M=2^p , where p = 1, 2, · · ·.
2. Initialize S=N (total number of nodes)
3. While S is greater than 0
4. Initialize X= 2^(floor(log2(S)))
5. Insert cluster of size X inside the array A
6. Subtract X from M
7. End
8. Return A

Each cluster is internally organized using the DADCNS algorithm as explained in section 3.2and 3.3. The algorithm for that design is given below.

Algorithm 2: (DADCNS Structure)

1. Inside each cluster, all members will be given a rank. The rank will be an integer value between 1 and p where p=2,3,4….. The rank will be allotted to nodes on the basis of various features such as distance to parent/base station, computing power, location etc.
2. A node with rank k will form k − 1 data links with k − 1 nodes, while these k − 1 nodes.All these k − 1 nodes will become the child nodes of the node with rank k.
3. This higher rank node will become the parent node of the node with rank k.
4. The cluster head is the one with the highest rank in the network and will have k maxusing equation 1 . The leaf nodes will have rank 1.
5. The Cluster head will connect to the base station.

NOTE: We do not have to explicitly check for collisions as it is impossible to happen in our system if all the base stations are accessed serially in the same order. This is because all the clusters in the network are of different sizes in the form of 2^p where p=1,2,3.. Therefore all cluster heads will take different no of timesots to send aggregated data to each BS. Hence, no two cluster heads will send data to the same base station at the same time slot.

Algorithm 3:- (MAIN ALGORITHM)

1. Initialize the value of |N| and K. Here, |N| denotes the total number of devices in an IoTnetwork and K is the total number of Base Stations.
2. Assign the nodes in multiple DADCNS clusters using Algorithm 1.
3. The nodes within the clusters will be arranged in DADCNS structure using Algorithm 2.
4. Calculate the time slots taken by each cluster to collect and aggregate all the data from allits nodes and forward the data to 1st Base Station using equation 3.
5. Store the time slots in a stack in an ascending order such that the cluster that is able to send the aggregated data to the 1st BS in the least required time slot is on the top and the one which requires the most time slots, in the bottom.
6. Create an array B of Base stations. The index of the array is how the base station is identified i.e. B0 , B1, B2, B3 …. BK. The clusters will access the BS serially in order.This will prevent multiple clusters from accessing the same BS in the same Time Slot.
7. The array of Base Stations will keep count of how many times each Base Station has been traversed by the clusters. All the base stations will be traversed an equal number oftimes(total number of clusters in the network) at the end.
8. Initialize a variable T which is the number of total time slots passed. Initially T will be 0.

9. Run Loop till all the Base Stations are visited G number of times. G being the totalnumber of clusters in the network.

10. If T == stack.top()

    Increase count in B0Then Pop stack.top()

    End.

    For i from 1 to K

    Increase count of B[i] by 1.

    That is, the cluster heads will access subsequent Base stations now.

    End.

    T=T+1

    If all the element in B is same i.e equal to the number of clustersBreak from Loop;

    End.

11. Return T(Total Time Slots)
12. Check if the resultant T (total time slot taken) matches with the result of Equation 4.

Let's consider a few examples:

Example 1: Let's consider a network with the number of Nodes N=29 and number of Base stations K=3.

Here the network will be divided into 4 clusters using Algorithm 1. The clusters will be of the size 16, 8, 4 and 1 using Algorithm 1. Each cluster will arrange themselves using Algorithm 2. Using equation 3 we can calculate the time required by each cluster to send aggregated data to the 1st base station. So for clusters of size 16 , 8, 4 and 1 , they are 5, 4, 3 and 1 respectively. Using equation 5 we can calculate the total timeslot required is floor(Log2(29)) + 3 = 7.

Example 2: Let's consider a network with the number of Nodes N=61 and number of Base stations K=7.

Here the network will be divided into 4 clusters using Algorithm 1. The clusters will be of the size 31, 16, 8, 4 and 1 using Algorithm 1. Each cluster will arrange themselves using Algorithm
2. Using equation 3 we can calculate the time required by each cluster to send aggregated data to the 1st base station. So for clusters of size 32, 16 , 8, 4 and 1 , they are 6, 5, 4, 3 and 1 respectively. Using equation 4 we can calculate the total timeslot required is floor(Log2(61)) + 7 = 12.

Example 3: Let's consider a network with the number of Nodes N=36 and number of Base stations K=4.

Here the network will be divided into 2 clusters using Algorithm 1. The clusters will be of the size 32 and 4 using Algorithm 1. Each cluster will arrange themselves using Algorithm 2. Using equation 3 we can calculate the time required by each cluster to send aggregated data to the 1st base station. So for clusters of size 32 and 4 , they are 6 and 3 respectively. Using equation 4 we can calculate the total timeslot required is floor(Log2(36)) + 4 = 9.

Chapter-4 PERFORMANCE ANALYSIS

The proposed model is evaluated and simulated in a custom-built python simulator. The results are then compared to Concurrent Data Collection Network Tree(CDCT) [2] and Time Optimal Concurrent Data Collection Tree(TOCDCT) [4]. These models are explained in chapter 2. The simulated results are then compared to see if the proposed model is actually faster than the two mentioned or not. The following sections will discuss the performance analysis further.

4.1 ALGORITHM ANALYSIS

Let's take an Example And analyze algorithms mentioned in section 3.4 accordingly:

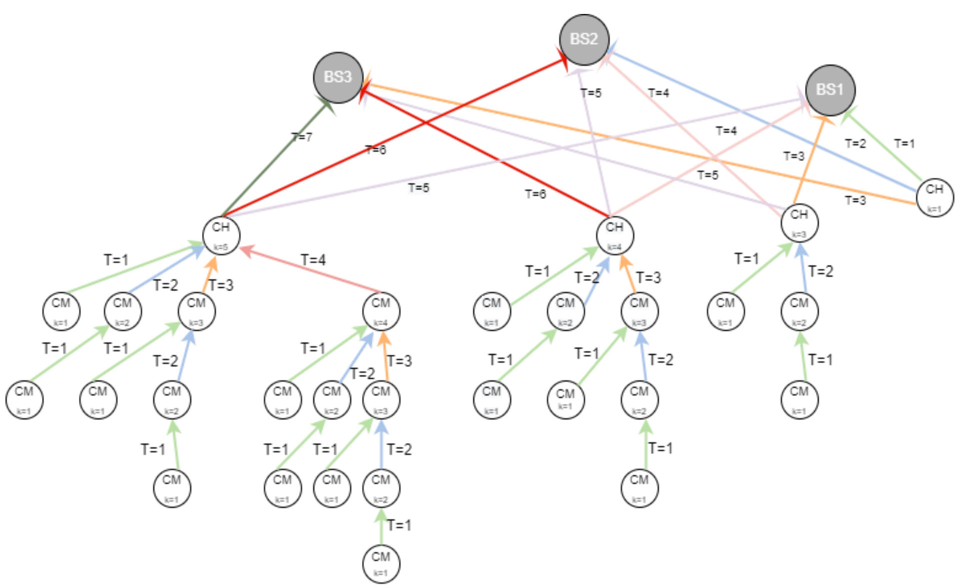


Fig. 4.1. Network size N = 29 for the proposed network structure. The cluster's members are shown as circles with CM. The cluster head is represented by a circle with CH. The base station is represented by a

filled circle with BS1,BS2,BS3 . The variable k represents the rank of each node. The existence of a data link is represented by a dashed arrow, and the direction of the arrow indicates the direction of data flow.

Consider a network with |N| = 29 and K = 3.(FIG 4.1)

1. The nodes will be divided into clusters of [16,8,4,1]
2. Each cluster will be internally arranged according to the DADCNS algo.
3. Then we calculate the kmax of each cluster head (which is equal to the timeslot in whichthe respective cluster head can forward collected data to the first BS).
4. Push the calculated values into the stack S.

   Stack S > TOP **[1 3 4 5]** BOTTOM
5. Create an array A for the count of times each base station is visited.

   Array A> [0 0 0]

   BS1 BS2 BS3
6. Run First Loop till the stack S is empty.

a. time_slot = 1

   S.top() == time_slotS.pop()

   // S > [3 4 5]A[0]+=1

   //A > [1 0 0]

b. time_slot = 2

   S.top() != time_slotA[1]+=1

   //A > [1 1 0]

c. time_slot = 3

   S.top() == time_slotS.pop()

   //S>[4 5]A[0]+=1

A[2]+=1

//A > [2 1   1]

d. time_slot = 4

S.top() == time_slotS.pop()

//S> [5]A[0]+=1A[1]+=1A[2]+=1

//A > [3 2   2]

e. time_slot = 5

S.top() == time_slotS.pop()

//S> [ ] A[0]+=1A[1]+=1A[2]+=1

//A > [4 3   3]

f. Stack S is empty. END LOOP.

7. Run Second Loop till Last cluster (kmax =5) has accessed all remaining BSs.

a. time_slot = 6

A[1]+=1

//A > [4 4   3]

b. time_slot = 7

A[2]+=1

//A > [4 4   4]

c. All the Base Stations are accessed an equal number of times (equal to number ofclusters). EXIT LOOP.

8. Return time_slot

The resultant time slot is found to be 7 which complies with the result from equation 4. The above mentioned example is tested on the model proposed in CDCT[2] and TOCDCT[4] and the results are found to be 8 and 8 respectively. So our model is performing the operation in lesser time slots.

4.2 SIMULATION RESULTS

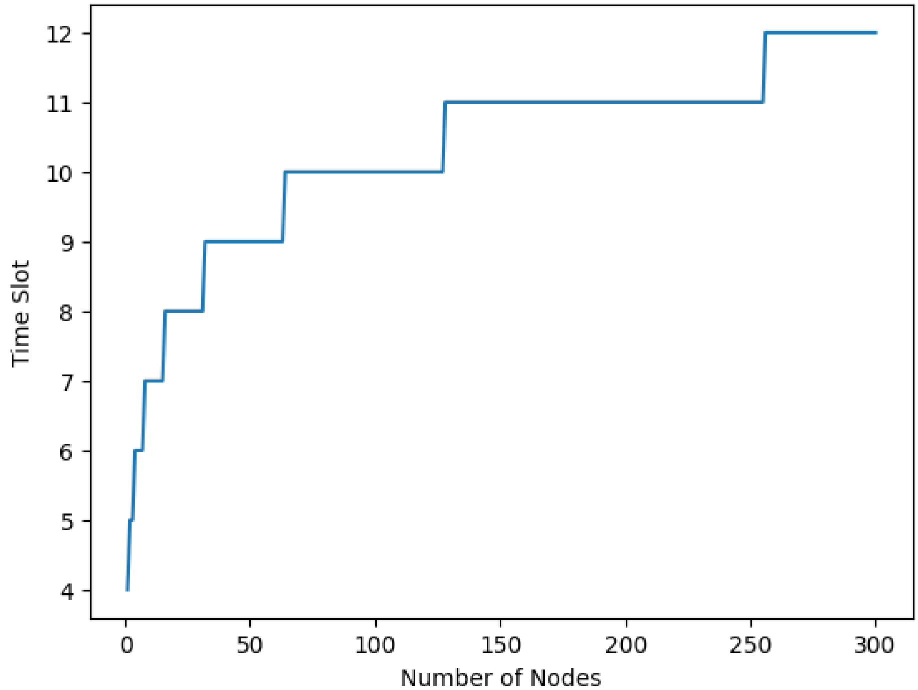
The simulations were performed in a python and c++ simulator that was created. The inputs of Nodes and base stations were varied and the results were then observed. The number of nodes
(N) were increased from 30 to 300 in a random fashion so that the inputs do not have any correlation with each other. Similarly the number of Base stations(K) were also increased from 1 to 10. The results were compared against the CDCT and TOCDCT model. The test was divided into two formats.

Since there are two factors, namely number of Nodes(N) and Base Stations(K) that are affecting our final result (Timeslots T), two cases were taken. Case I would have a gradual increase of the number of Nodes from 30 to 300 while keeping the number base stations(K) constant. Case II will have an increase of Base Station(K) from 1 to 10 while keeping the number nodes constant. We took three different scenarios in each case. The results are displayed in both table and bar chart format.

```
Enter the number of Nodes = 300
Enter the number of Base Stations = 3
The cluster head created are as follows: -
 256 32 8 4
Cluster Head CH1: -
The Cluster head CH1 transmitted data to base station B1 at timeslot 3.
The Cluster head CH1 transmitted data to base station B2 at timeslot 4.
The Cluster head CH1 transmitted data to base station B3 at timeslot 5.

Cluster Head CH2: -
The Cluster head CH2 transmitted data to base station B1 at timeslot 4.
The Cluster head CH2 transmitted data to base station B2 at timeslot 5.
The Cluster head CH2 transmitted data to base station B3 at timeslot 6.

Cluster Head CH3: -
The Cluster head CH3 transmitted data to base station B1 at timeslot 6.
The Cluster head CH3 transmitted data to base station B2 at timeslot 7.
The Cluster head CH3 transmitted data to base station B3 at timeslot 8.

Cluster Head CH4: -
The Cluster head CH4 transmitted data to base station B1 at timeslot 9.
The Cluster head CH4 transmitted data to base station B2 at timeslot 10.
The Cluster head CH4 transmitted data to base station B3 at timeslot 11.
```

Fig. 4.2. Network size N = 300 and base station = 3 and respective time taken by each cluster head to transmit data from cluster head to base station.

4.2.1 CASE I : Keeping the number of Base Stations(K) constant



Graph 4.1 Here keeping the base station constant = 4 and number of node vary from 0 to 300.

I. Scenario 1:

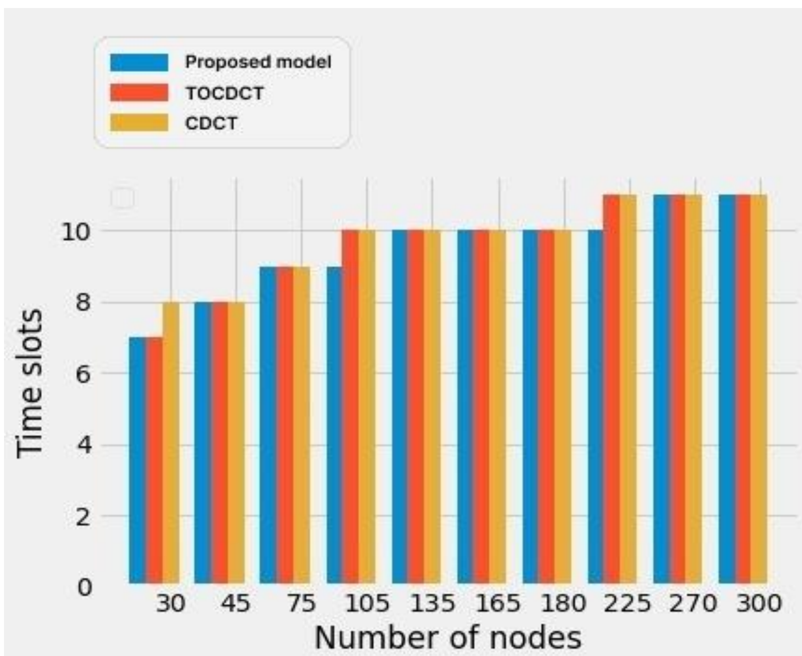
Let's consider a case where the number of Base Stations K = 3 is kept constant and the number for nodes is increased from 30 to 300. The time slots of CDCT, TOCDCT and our proposed model are compared.

TABLE 4.1

The number of Nodes increased from 30 to 300 while keeping the number base stations(K) constant at K=3.

No. of Nodes (N)	No. of Base Stations (K)	Time CDCT	Time TO-CDCT	Time Proposed Model
30	3	8	7	7
45	3	8	8	8
75	3	9	9	9

105	**3**	**10**	**10**	**9**
135	**3**	**10**	**10**	**10**
165	**3**	**10**	**10**	**10**
180	**3**	**10**	**10**	**10**
225	**3**	**11**	**11**	**10**
270	**3**	**11**	**11**	**11**
300	**3**	**11**	**11**	**11**



Graph 4.2 The number of Nodes increased from 30 to 300 while keeping the number base stations(K) constant at K=3.

Observations:

As observed from Graph 4.1, the timeslots of all three models are very much similar for K=3. Although for N=30, our proposed model and Time Optimal CDCT performs better than CDCT. For N=105 and N=225 our model performs slightly better than both CDCT And Time Optimal CDCT. So for a smaller number of base stations, all three model perform very much equally with our proposed model being just a little bit faster.

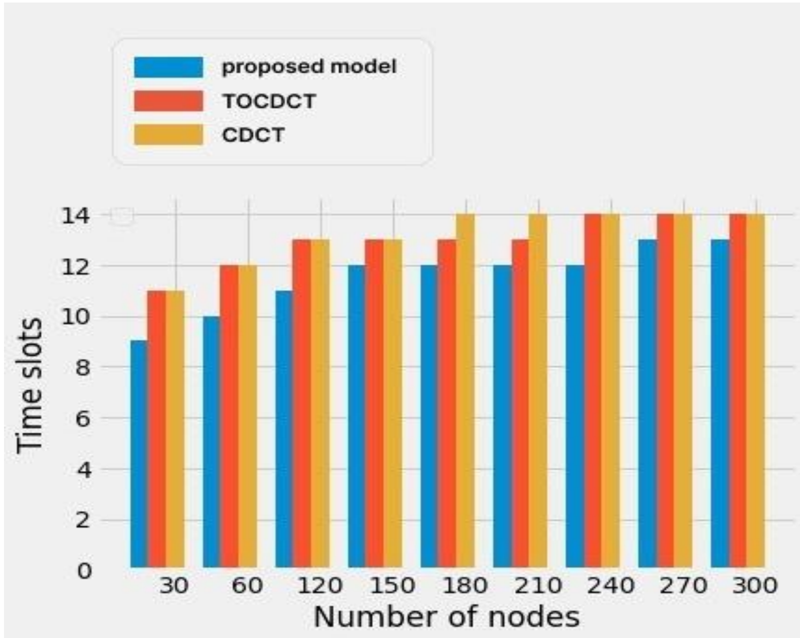
II. Scenario 2:

Let's consider a case where the number of Base Stations K = 5 is kept constant and the number for nodes is increased from 30 to 300. The time slots of CDCT, TOCDCT and our proposed model are compared.

TABLE 4.2

The number of Nodes increased from 30 to 300 while keeping the number base stations(K) constant at K=5.

No. of nodes (N)	No. of Base stations (K)	Time CDCT	Time TO-CDCT	Time Proposed Model
30	5	11	11	9
60	5	12	12	10
120	5	13	13	11
150	5	13	13	12
180	5	14	13	12
210	5	14	13	12
240	5	14	14	12
270	5	14	14	13
300	5	14	14	13



Graph 4.3 The number of Nodes increased from 30 to 300 while keeping the number base stations(K) constant at K=5.

Observations:

As observed from Graph 4.2, the timeslots of our proposed model seem to be better than CDCT and TOCDCT. Our model performed better in all the cases. For low and high numbers of Nodes, CDCT and TO CDCT seem to provide similar results, while our proposed model being faster in all the test cases.

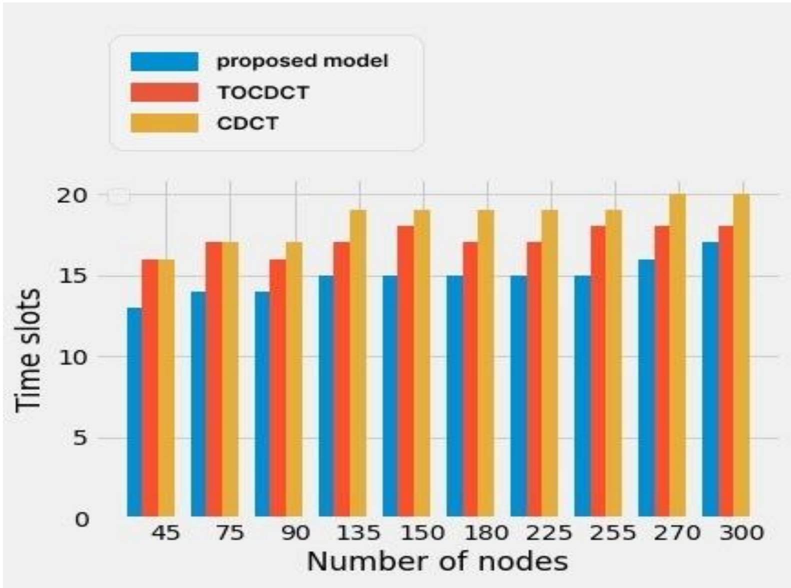
III.Scenario 3:

Let's consider a case where the number of Base Stations K = 8 is kept constant and the number for nodes is increased from 30 to 300. The time slots of CDCT, TOCDCT and our proposed model are compared.

TABLE 4.3

The number of Nodes increased from 30 to 300 while keeping the number base stations(K) constant at K=8.

No. of Nodes (N)	No. of Base Stations (K)	Time CDCT	Time TOCDCT	Time Proposed Model
45	8	16	16	13
75	8	17	17	14
90	8	17	16	14
135	8	19	17	15
150	8	19	18	15
180	8	19	17	15
225	8	19	17	15
255	8	19	18	15
270	8	20	18	16
300	8	20	18	16

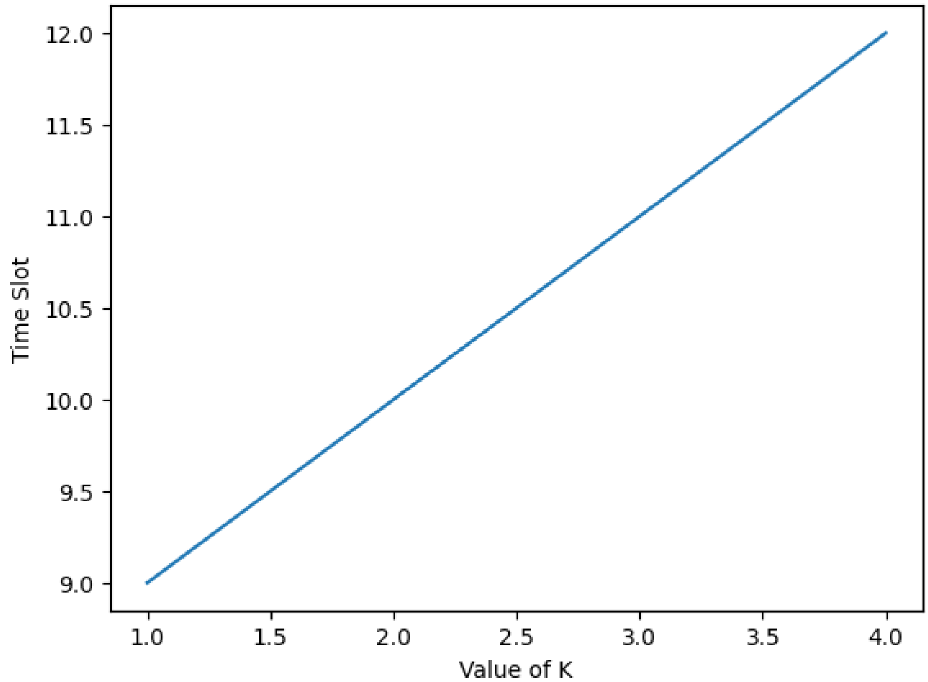


Graph 4.4 The number of Nodes increased from 30 to 300 while keeping the number base stations(K) constant at K=8.

Observations:

As observed from Graph 4.3, the timeslots of our proposed model seem to be better than CDCT and TOCDCT. Our model performed better in all the cases. For a low number of Nodes, CDCT and TO CDCT seem to provide similar results, while our proposed model being faster in all the test cases. For a higher number of nodes both our model and TO CDCT performs better than CDCT. We can conclude that as the number of Nodes get higher our models also perform better than CDCT and TO CDCT for the same number of base stations.

4.2.1 CASE I : Keeping the number of Nodes(N) constant



Graph 4.5 Keeping the number of node constant = 300 and number of base station = 4 that is varying from 1 to 4.

I. Scenario 1:

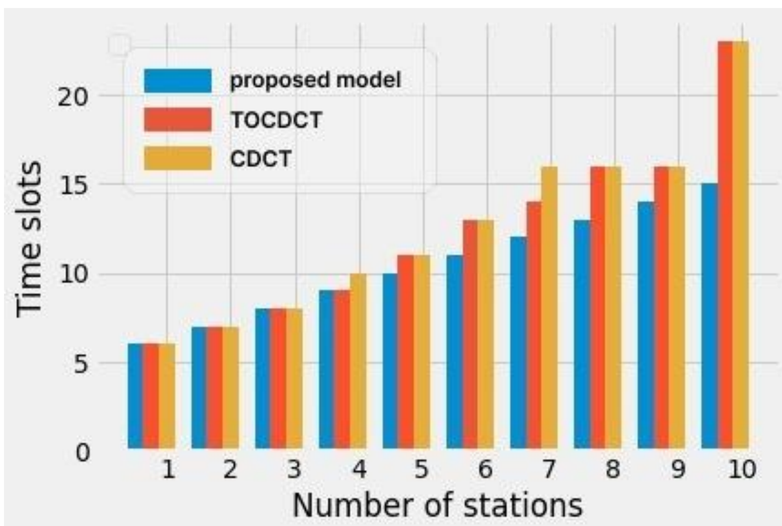
Let's consider a case where the number of Nodes N = 45 is kept constant and the number for basestations K is increased from 1 to 10. The resulting time slots of CDCT, TO CDCT and our proposed model are compared.

TABLE 4.4

The number of Base Station(K) increased from 1 to 10 while keeping the number of Nodes(N) constantat N=45.

No. of Nodes (N)	No. of Base Stations (K)	Time CDCT	Time(T) TO-CDCT	Time(T) Proposed Model
45	**1**	**6**	**6**	**6**
45	**2**	**7**	**7**	**7**

45	**3**	**8**	**8**	**8**
45	**4**	**10**	**9**	**9**
45	**5**	**11**	**11**	**10**
45	**6**	**13**	**13**	**11**
45	**7**	**16**	**14**	**12**
45	**8**	**16**	**16**	**13**
45	**9**	**16**	**16**	**14**
45	**10**	**23**	**23**	**15**



Graph 4.5 The number of Base Station(K) increased from 1 to 10 while keeping the number ofNodes(N) constant at N=45.

Observations:

As observed from Graph 4.4, the timeslots of the proposed model seem to be better than CDCT and TOCDCT. Our model performed better in all the cases. For a low number of Base Stations, our model, CDCT and TO CDCT seem to provide similar results, For a higher number TO CDCT performs better than CDCT while the proposed model performs better in all the cases. The interesting case is when K=10. The proposed model seems to perform much better than CDCT and TO CDCT. This might be due to the case that the proposed model structure is better suited to the number of Nodes and Base stations in this case while the ring structures of CDCT and TO CDCT can converge the data between all nodes and base stations as quickly.

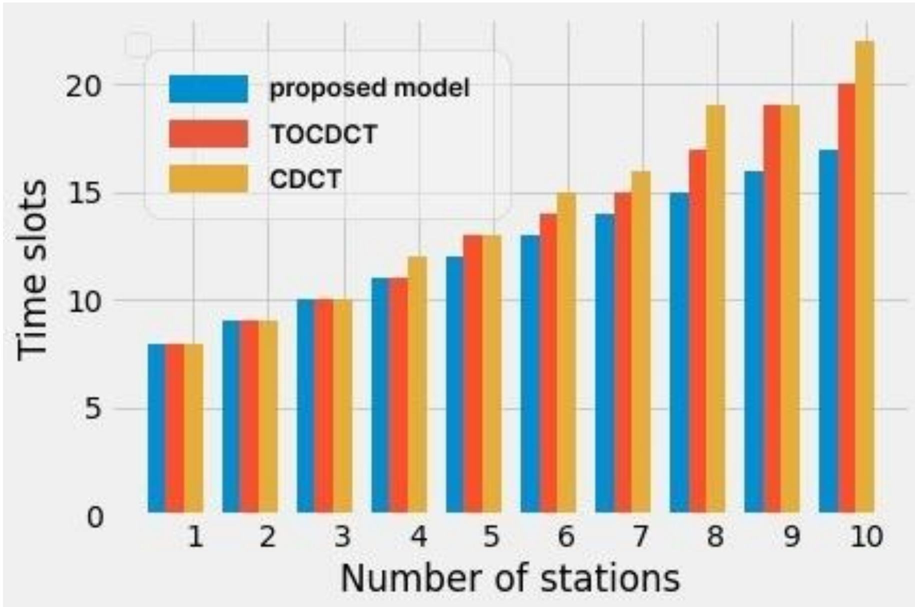
II.Scenario 2:

Let's consider a case where the number of Nodes N = 135 is kept constant and the number for base stations K is increased from 1 to 10. The resulting time slots of CDCT, TO CDCT and our proposed model are compared.

TABLE 4.5

The number of Base Station(K) increased from 1 to 10 while keeping the number of Nodes(N) constantat N=135.

No. of Nodes (N)	No. of Base Stations (K)	Time CDCT	Time TOCDCT	Time Proposed Model
135	1	8	8	8
135	2	9	9	9
135	3	10	10	10
135	4	12	11	11
135	5	13	13	12
135	6	15	14	13
135	7	16	15	14
135	8	19	17	15
135	9	19	19	16
135	10	22	20	17



Graph 4.6 The number of Base Station(K) increased from 1 to 10 while keeping the number ofNodes(N) constant at N=135.

Observations:

As observed from Graph 4.5, the timeslots of the proposed model seem to be better than CDCT and TOCDCT. Our model performed better in all the cases. For a low number of Base Stations, our model, CDCT and TO CDCT seem to provide similar results, For a higher number TO CDCT performs better than CDCT while the proposed model performs better in all the cases. Wecan see that the proposed model's performance gets better and better as the number of base stations K increases.

III.Scenario 3:

Let's consider a case where the number of Nodes N = 255 is kept constant and the number for base stations K is increased from 1 to 10. The resulting time slots of CDCT, TO CDCT and our proposed model are compared.

TABLE 4.6

The number of Base Station(K) increased from 1 to 10 while keeping the number of Nodes(N) constantat N=255.

No. of Nodes (N)	No. of Base Stations (K)	Time CDCT	Time TOCDCT	Time(T) Proposed Model
255	1	8	8	8
255	2	9	9	9
255	3	11	11	10
255	4	12	12	11
255	5	14	14	12
255	6	16	15	13
255	7	18	17	14
255	8	19	18	15
255	9	22	19	16
255	10	23	21	17

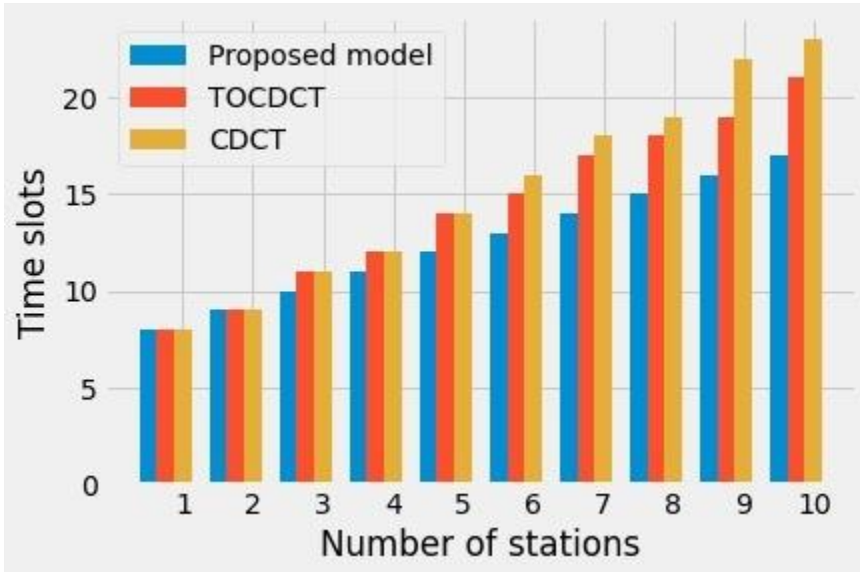


Fig Graph 4.7 The number of Base Station(K) increased from 1 to 10 while keeping the number of Nodes(N) constant at N=255.

Observations:

As observed from Graph 4.6, the timeslots of the proposed model seem to be better than CDCT and TOCDCT. Our model performed better in all the cases. Similar to scenario 2, for a low number of Base Stations, our model, CDCT and TO CDCT seem to provide similar results, For ahigher number TO CDCT performs much better than CDCT while the proposed model performs better in all the cases. We can see that the proposed model's performance gets better and better asthe number of base stations K increases.

4.2 RESULT DISCUSSION

From the experiments performed in section 4.2, it is observed that for all cases the proposed model performs better than both Concurrent Data Collection Tree(CDCT) and Time Optimal CDCT. It is also seen that as the number of nodes (N) and base stations(K) are increased, the resulting timeslots from the proposed network structure performs even better compared to the other two models.

The simulation results support the findings of our analysis, which were covered in the previous section. The total number of time-slots required by the network using the proposed time optimal data collection trees is significantly less when compared to networks using CDCT and Time Optimal CDCT.

When |N| = 150 and k = 7, for example, data collection takes place over 17 time slots in CDCT and 16 time slots in Time Optimal CDCT, whereas it takes place over 14 time slots in the proposed approach for the same values of |N| and k.

Similar to this, when |N| = 250 and k = 10, the duration of data collection in CDCT and Time Optimal CDCT is 23 timeslots and 20 timeslots, respectively, but the proposed approach's duration is 17 timeslots for the same values of |N| and k.

The gains of the suggested network structures increase as |N| and K are increased, and it is demonstrated that increase in clusters causes the number of time slots T to increase slowly. The number of clusters will rise proportionally as |N| increases, and as more devices participate in concurrent transmissions, the proposed tree structure's total number of slots T will be lower than that of CDCT and Time Optimal CDCT.

Chapter-5 CONCLUSIONS

5.1 CONCLUSIONS

In this study, we suggest the ideal technique for IoT systems to collect data concurrently. Due to the enhanced advantages of shared device infrastructure, an increasing number of Internet of Things (IoT) applications rely on such shared systems for their data needs. Concurrent data delivery is required in order to maintain the freshness of data as multiple such applications request data at once. Furthermore, in real-time systems, the timely delivery of data is essential for making critical decisions. Here, a combination of network topologies is used in the proposed network structure to reduce the delays. The results of the simulation and performance analysis demonstrate that the proposed process outperforms the currently used data collection methods, CDCT and Time Optimal CDCT.

It is predicted that public and private internet of things (IoT) systems will soon be connected to create an IoT federation. IoT devices will be shared among various parties under these connected systems. On the same collection of IoT devices, various data collection processes started by various users can run concurrently.

Specifically created for concurrent data collection processes in IoT systems. The suggested network structure can reduce the delays caused by multiple data collection processes running at once. According to the findings of this project, the suggested idea can result in shorter data collection times than a network structure that is currently in use and was created for a single data collection process. Also provided are comprehensive instructions for obtaining workable transmission schedules for the suggested network structure.

Network construction can be done centrally or decentralizedly to fit diverse purposes. Two network creation techniques are developed to achieve the best results for networks of various sizes. A multiple-cluster 2-hop network structure, a single-chain network structure, a minimal spanning tree network structure, and a collection tree network structure are used to evaluate the performance of the proposed network structure. Among the network designs listed above, the proposed network structure is shown to be the most efficient in terms of data collection time. The proposed network layout can greatly reduce data gathering time while keeping overall communication distance and network lifetime within tolerable constraints.

5.2FUTURE SCOPE

The Internet of Things has become a dominant technology globally. In a short period of time, it has become extremely popular. In addition, the automation of IoT devices has become simple thanks to advancements in artificial intelligence and machine learning. In essence, IoT devices are combined with AI and ML programmes to properly automate them. As a result, the IoT has broadened the range of industries in which it can be applied. We'll talk about the uses and potential of IoT in the healthcare, automotive, and agricultural industries in this section.

It is simple to adapt the process for determining the transmission schedules in the proposed network configuration to accommodate for additional optimization limitations or criteria. Mobile networks frequently worry about the entire communication distance of the data gathering tree since it could shorten the battery life of mobile devices. The N2N communication distance between each member of a cluster of the proposed structure can be reduced by using clustering algorithms with predetermined cluster sizes. This value can be further decreased and the length of the ring's total path can be shortened by employing travelling salesman problem solvers to rearrange the nodes inside each loop. Additional variables, such as channel quality and bandwidth, can also be included to transform the methods into a multi-objective optimization process. Concurrent transmission interferences are a different problem, but they can be avoided or at least minimised by putting minimal separation requirements on conflicting nodes when making practical transmission schedules. Additionally, by using multiple communication channels, which is a practical choice for the majority of modern transceiver modules, it is possible to lessen interference between IoT devices. Fast data collection from IoT sensors can help receive data in lesser delay form ever before. which can eventually help benefit globally since IoT has become a day to day application in our lives. This project could be extended to various domains. For this particular research, we have only looked at the structure of the network between the nodes and the base stations. In future we can also look into the frequency of each device and the energy consumption of these data collection processes and try to minimize it. This could in turn help us to reduce the use of energy to operate an IoT network.

By processing big data, cloud computing is currently generating a lot of interest across many industries. where information is gathered from a variety of sources, including sensor networks, social networks, and automobiles. Concerns about the security of the data transferred to the cloud data center from the aforementioned sources can still be addressed. To support the data collection from sensors to the cloud, a common architecture is required.

Any communication system that adheres to a specific Quality of Service for each communication application is built on a foundation of network protocols. Remote system architecture becomes increasingly capable as installed technology advances quickly, and its topological structure and correspondence become more unpredictable. The proposed model could be extended to other data collection domains as well including data collection from WSNs etc.

5.3APPLICATIONS

Everyone wants a job done for them without any effort as the world's population shifts toward relying more on technology than manual methods. Internet of Things essentially refers to computing devices that transmit and receive data over the internet. Considering its advantages and the degree of comfort people are experiencing, IoT is becoming a significant part of their lives. IoT can assist humanity in many ways, some of which are listed below:

1. The medical industry: Extensive implementation is possible. Checkups, wearable health devices, telemedicine, and many other things.

2. Smart Homes: Several devices, including Google Home, Amazon's Alexa, and Nest, have been introduced. Each of these devices has a specific function that makes it possible for members of a household to communicate with one another online and improves the quality of our lives.

3. Smart Cities: Time-wasting traffic jams are the major issue in large cities, but IoT is facilitating connectivity and information sharing to allow for proactive situational management. security systems, cutting-edge parking systems.

There are numerous additional industries, including manufacturing, advanced power supply, planning, industrial automation, and the digitization of cities in developing nations (see, for instance, Mark Zuckerberg's JARVIS). There are countless opportunities.

References

[1] C.-T. Cheng, K. T. Chi, and F. C. Lau, “A delay-aware data collection network structure for wireless sensor networks,” IEEE sensors journal, vol. 11, no. 3, pp. 699–710, Mar. 2011.

[2] C. Cheng, N. Ganganath, and K. Fok, “Concurrent data collection trees for IoT applications,” IEEE Transactions on Industrial Informatics, vol. 13, no. 2, pp. 793–799, Apr. 2017.

[3] Ozlem D Incel, Amitabha Ghosh, and Bhaskar Krishnamachari, “Fast Data Collection in Tree-Based Wireless Sensor Networks” IEEE Transactions on Mobile Computing 11(1),February 2012

[4] Arvind Kumar, Rakesh Matam, and Mithun Mukherjee, “Time Optimal Concurrent Data collection Trees for IoT Applications” IEEE International Systems Conference(SysCon), April 2021

[5] J. N. Al-karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” IEEE Wireless Communications Mag., vol. 11, no. 6, pp. 6–28, December 2004.

[6] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Trans. Wireless Communications, vol. 1, no. 4, pp. 660–670, October 2002

[7] S. Lindsey and C. S. Raghavendra, “PEGASIS: Power-efficient gathering in sensor information systems,” in Proc. IEEE Conf. Aerospace, vol. 3, Big Sky, Montana, USA, March 2002, pp. 1125–1130.

[8] H. O. Tan and ¨ ´I. Korpeo ¨ glu, “Power efficient data gathering and ag- ˆ gregation in wireless sensor networks,” ACM SIGMOD Record, vol. 32, no. 4, pp. 66–71, December 2003.

[9] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo, “The collection tree protocol,” TinyOS Enhancement Proposals (TEP) 123, December 2007.

[10] C. Florens, M. Franceschetti, and R. J. McEliece, “Lower bounds on data collection time in sensory networks,” IEEE Jour. Selected Areas in Communications, vol. 22, no. 6, pp. 1110–1120, August 2004.

[11]I. Solis and K. Obraczka, “The impact of timing in data aggregation for sensor networks,” in IEEE International Conf. on Communications (IEEE Cat. No. 04CH37577), vol. 6, 2004, pp. 3640–3645.

[12]M. Song and B. He, “Capacity analysis for flat and clustered wireless sensor networks,” in Proc. Int. Conf. Wireless Algorithms, Systems and Applications, (WASA 2007), Chicago, Illinois, USA, August 2007, pp. 249–253.

[13]V. Annamalai, S.K.S. Gupta, L. Schwiebert, “On tree-based convergecasting in wireless sensor networks”, in WCNC ’03, pp.

1942–1947.

[14]S. Gandham, Y. Zhang and Q. Huang, “Distributed Time-Optimal Scheduling for Convergecast in Wireless Sensor Networks”, Computer Networks, vol. 52, nr. 3, 2008, pp. 610–629.

[15]T. Moscibroda, “The worst-case capacity of wireless sensor networks”, in IPSN ’07, pp. 1–10

[16]G. Chen, J. Tang, and J. P. Coon, “Optimal routing for multihop socialbased d2d communications in the internet of things,” IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1880–1889, 2018.

[17]O. Tekdas, J. H. Lim, A. Terzis, and V. Isler, “Using mobile robots to harvest data from sensor fields,” IEEE Wireless Communications Mag., vol. 16, no. 1, pp. 22–28, February 2009

[18]W. Zhao and X. Tang, “Scheduling sensor data collection with dynamic traffic patterns,” Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 4, pp. 789–802, 2013.

[19]S. Ji, R. Beyah, and Z. Cai, “Snapshot and continuous data collection in probabilistic wireless sensor networks,” Mobile Computing, IEEE Transactions on, vol. 13, no. 3, pp.626–637, 2014.

Z. Cai, S. Ji, J. He, L. Wei, and A. Bourgeois, “Distributed and

[20]asynchronous data collection in cognitive radio networks with fairness consideration," Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 8, pp. 2020–2029, 2014.

[21] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," Sensors Journal, IEEE, vol. 13, no. 10, pp. 3558–3567, 2013

[22] R. Kolcun, D. Boyle, and J. A. McCann, "Optimal processing node discovery algorithm for distributed computing in IoT," in Proc. 5th International Conf. on the Internet of Things (IoT), 2015, pp. 72–79

APPENDICES

Python 3 simulator used for this project:

Node is incremented while base station is kept constant

```
import math
import matplotlib.pyplot as plt
N=int(input("Enter number of Nodes = "))
k=int(input("Enter number of Base Stations = "))
mx=[]
timing=[]
for m in range(1,N+1):
  mx.append(m)
  li=[]
  while m>0:
    x=math.pow(2,math.floor(math.log2(m)))
    li.append(x)
    m-=x
  tm=[]
  for i in range(0,len(li)):
    y=(math.log2(li[i]))+1
    tm.append(y)
  arr=[]
  for i in range(0,k):
    arr.append(0)
  st = []
  for i in range(0,len(tm)):
    st.append(tm[i])
  time=0
  while st:
    time+=1
    for i in range(1,k):
      if arr[i]<arr[i-1]:
        arr[i]+=1
      else:
        break
    if st[-1]==time:
      arr[0]+=1
      st.pop()
```

```
  for i in range(1,k):
    arr[i]+=1
    time+=1
  timing.append(time)

print(mx)
print(timing)
plt.plot(mx,timing)
plt.xlabel("Number of Nodes")
plt.ylabel("Time Slot")
plt.show()
```

```
# Base station is incremented while Node is kept

constant

import math
import matplotlib.pyplot as plt
N=int(input("Enter number of Nodes = "))
K=int(input("Enter number of Base Stations = "))mx=[]
timing=[]
for k in range(1,K+1):mx.append(k)
     m=N li=[]
     while m>0:
     x=math.pow(2,math.floor(math.log2(m)))li.append(x)
         m-=xtm=[]

 for i in range(0,len(li)):y=(math.log2(li[i]))+1
  tm.append(y)

     arr=[]
 for i in range(0,k):arr.append(0)

     st = []
 for i in range(0,len(tm)):st.append(tm[i])

     time=0 while st:
     time+=1
      for i in range(1,k):
 if arr[i]<arr[i-1]:arr[i]+=1
     else:
     break
 if st[-1]==time:arr[0]+=1 st.pop()

 for i in range(1,k):arr[i]+=1 time+=1
     timing.append(time)

     print(mx) print(timing)

plt.plot(mx,timing)
plt.xlabel("Value of K")
plt.ylabel("Time Slot")
plt.show()
```

```
# Data transmission between cluster head and Base
station.

     import math

     basedi={}
     n=int(input("Enter the number of Nodes = "))

k=int(input("Enter the number of Base Stations = "))
     nli=[]
     while n>0:
         a=math.floor(math.log2(n))
         ch=pow(2,a)
         b=a+1
         basedi[ch]=b
         n=n-ch
         nli.append(ch)


     print("The cluster head created are as follows: -
\n",*nli)
     z=1
     for i in range(len(nli)-1,-1,-1):
         a=basedi[nli[i]]
         print("Cluster Head CH{0}: -".format(z))
         for j in range(1,k+1):

print("The Cluster head CH{0} transmitted data to base s
    tation B{1} at times        {2}.".format(z,j,a))
             a+=1
         print()
         z+=1
```

CPP simulator Code :

```
#include<bits/stdc++.h>using namespace std; int main() {
// Write C++ code hereint N,k;
cout<<"Enter number of Nodes: ";cin>>N;
cout<<"Enter number of Base Stations: ";cin>>k;
//breaking the nodes into clusters of 2^p where
p=1,2,3..vector<int> v;
int M=N; while(M>0)
{
int x=pow(2,floor(log2(M)));v.push_back(x);
M-=x;
}
cout<<"The clusters will be of size each :"<<endl;
for(int i=0;i<v.size();i++)
cout<<v[i]<<" ";
    //calculating time taken by each cluster head to
forward aggregateddata to the 1st BS
cout<<endl; vector<int> tm;
for(int i=0;i<v.size();i++)
{
int k=log2(v[i])+1;tm.push_back(k);
}
    cout<<"timeslots taken by each Cluster Head(CH) to
forward aggregateddata to the 1st BS"<<endl;
for(int i=0;i<tm.size();i++)cout<<tm[i]<<" ";

stack<int> st;
for(int i=0;i<tm.size();i++)
```

```
{
st.push(tm[i]);
}
int time=0;int arr[k];
for(int i=0;i<k;i++)
{
arr[i]=0;
}
cout<<endl;
    cout<<"Intially each base station is visited 0 times
as shown:"<<endl;
for(int i=0;i<k;i++)
{
cout<<arr[i]<<" ";
}
while(!st.empty())
{
time+=1;
for(int i=1;i<k;i++)
{
if(arr[i]<arr[i-1])
{
arr[i]+=1;
}
else
break;
}
if(st.top()==time)
{
arr[0]+=1;st.pop();
}
}
for(int i=1;i<k;i++)
{
arr[i]+=1;time+=1;
}
cout<<endl;
```

```
    cout<<"All the base stations are visited by all the clusters(whosecount is equal to the number of clusters) as shown "<<endl;
for(int i=0;i<k;i++)
{
cout<<arr[i]<<" ";
}
cout<<endl;
cout<<"Total time slots taken: "<<time;return 0;
}
```