

RANDOMIZED GENERATION TECHNIQUE FOR SECURED DATA TRANSMISSION BY CRYPTOGRAPHIC MECHANISM

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree

in

Electronics and Communication Engineering

By

PATHAKAMURI SRINIVAS (37130302)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PATHAKAMURI SRINIVAS (REG.NO 37130302)** who carried out the project entitled **"RANDOMIZED GENERATION TECHNIQUE FOR SECURE DATA TRANSMISSION BY CRYPTOGRAPHIC MECHANISM"** under my supervision from AUG 2020 to MARCH 2021.

Internal Guide

Ms. S. YOGALAKSHMI, M.E., (Ph.D).,

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I PATHAKAMURI SRINIVAS (37130302) hereby declare that the Project Report entitled “RANDOMIZED GENERATION TECHNIQUE FOR SECURED DATA TRANSMISSION BY CRYPTOGRAPHIC MECHANISM” done by us under the guidance of Ms. S. YOGALAKSHMI, M.E., (Ph.D)., is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1.

DATE : 17.04.2021

PLACE : CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to **Dr. N.M. NANDITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. S. YOGALAKSHMI, M.E., (Ph.D.)**, Asst. Professor, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

This article explores many issues relating to the Wireless Sensor Network (WSN) and the relevance of Elliptic Curve Cryptography (ECC). WSN security is more noteworthy in WSN, given that sensor hubs and remote connections are impeded by preparation. Widespread use of WSNs leads to different kinds of hazards. Adequate protection plans are required to protect against the dangers. Customary authentication is done by machinery or programming and generally carried out by means of encryption techniques. Limited area, existence of connections, restricted handling, force and WSN memory imply exact restrictions on cryptographic method determination. This paper looks at the data transfer over IOT and the two modified RSA and elliptical curve cryptography (ECC) cryptography calculations are the best applicants for their smaller scale. Despite the smaller key size, high security creates regional and efficient crypt structures.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	V
	LIST OF FIGURES	Ix
	LIST OF ABBREVIATIONS	Xi
1.	INTRODUCTION	1
1.1	METHOD OF APPROACHING	2
2.	LITERATURE SURVEY	5
2.1	COMPARISON OF LITERATURE SURVEY	9
3.	AIM AND SCOPE	10
3.1	EXISTING SYSTEM	10
3.2	PROPOSED SYSTEM	11
3.3	NEIGHBOR DISCOVERY BLOCK	12
3.4	BROADCASTING NODES	13
4.	ALGORITHM AND IMPLEMENTATION	14
4.1	ALGORITHMS	14
4.1.1	ECC ALGORITHM (ELLIPTICAL CURVE CRYPTOGRAPHY)	14
4.1.2	RSA ALGORITHM (RIVEST SHAMIR ADLEMAN)	17
4.2	IMPLEMENTATION	19
4.2.1	OTCL ESSENTIALS	20
4.2.2	ASSIGNING ESTEEMS TO FACTORS	20
4.2.3	PROCEDURES	21
4.2.4	FILES AND RECORDS	22

CHAPTER NO.	TITLE	PAGE NO.
	4.2.5 CALLING SUBPROCESSES	22
	4.2.6 CREATING THE GEOGRAPHY	23
	4.2.7 NODES	23
	4.2.8 AGENTS, APPLICATIONS AND TRAFFIC SOURCES	24
	4.2.9 TRAFFIC SINKS	25
	4.2.10 LINKS	26
	4.2.11 TRACING AND CHECKING	28
	4.2.12 TRACES	28
	4.2.13 MONITORS	30
	4.2.14 CONTROLLING THE RECREATION	30
	4.3 SIMPLE STIMULATION EXAMPLE	31
	4.4 EXTENDING NS: CREATING ANOTHER AGENT	37
	4.5 THE HEADER RECORD	37
	4.5.1 TABLE OF MYPING.H	40
	4.5.2 TABLE OF MYPING.CC	44
	4.6 NECESSARY CHANGES IN NS	44
5.	RESULTS AND DISCUSSION	45
	5.1 EDIT PACKET.H FILE	45
	5.2 EDIT NS-PACKET.TCL DOCUMENT	46
	5.3 EDIT NS-DEFAULT.TCL	47
	5.4 EDIT MAKEFILE.IN	47
	5.5 RECOMPILE NS	48
	5.6 SIMPLE EXAMPLE USING MYPING AGENT	48
	5.7 RSA – NAM RESULTS	49
	5.8 ECC – NAM RESULTS	51
	5.9 PARAMETER ANALYSIS	53

CHAPTER NO.	TITLE	PAGE NO.
	5.9.1 RSA RESULTS	53
	5.9.2 ECC RESULTS	55
6.	SUMMARY AND CONCLUSION	59
	REFERENCES	60

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
4.1	FORWARD ROUTE DISCOVERY (ECC)	16
4.2	BACKWARD ROUTE DISCOVERY (ECC)	17
4.3	KEY GENERATION IN ECC	17
4.4	LINK IN NS2	27
4.5	LINK IN NS2 WHEN TRACING IS ENABLED	29
4.6	A SIMPLE NETWORK TOPOLOGY AND STIMULATION SCENARIO	32
4.7	NS DIRECTORY STRUCTURE	44
5.1	NETWORK REPRESENTATION NAM OUTPUT FILE (RSA)	49
5.2	AUTHENTICATION BETWEEN NODES	49
5.3	VALIDING SIGNATURE (KEYS) IN NETWORK	50
5.4	ROUTING IN NETWORK OVER OFF THE ATTACKERS	50
5.5	RECEIVING DATA AT RECEIVER NODE FROM TRANSMITTER NODE	51
5.6	NETWORK REPRESENTATION NAM OUTPUT (ECC)	51
5.7	DATA TRANSMISSION BETWEEN TRANSMITTER AND RECEIVER NODES	52
5.8	ROUTING PATH BETWEEN SOURCE AND DESTINATION	52
5.9	PACKET DELIVERY RATIO	53
5.10	PACKET LOSS RATIO	53
5.11	ROUTING OVERHEAD	54
5.12	COMPARISON OF ENERGY	54
5.13	END TO END DELAY	55
5.14	NETWORK PERFORMANCE OF ECC	55
5.15	PACKET LOSS RATIO	56

FIGURE NO.	TITLE	PAGE NO.
5.16	PACKET DELIVERY RATIO	56
5.17	COMPARISON OF ENERGY	57
5.18	ROUTING OVERHEAD	57
5.19	END TO END DELAY	58

LIST OF ABBREVIATIONS

BPA	-	Basic Probability Assigning
ECC	-	Elliptical Curve Cryptography
FIFO	-	First-In First-Out
MR	-	Magnetic Resonance
QoS	-	Service Quality
RSA	-	Rivest Shamir Adleman
WSAN	-	Wireless Sensors and Actuator
WSN	-	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

Today, with the rapid advancement of communication domains, instant delivery and arrival of data over a network or different networks is possible. But along with it much emphasis is given on the security of data during its traversal over the network. Data can be identified, modified and even destroyed by unauthorized users during its traversal. So, the solution to this is to transmit encrypted data over the network so that it is impossible to identify the message by the hands of the unauthorized users.

Cryptography is an art and science of mathematically scrambling the data so that only someone with the necessary key can unscramble it. It contains mainly two domains i.e. Encryption and Decryption. Web of stuff (IoT) is a first invention that focuses on the connecting of stuff or gadgets and people or customers to achieve such common aims. IoT is powered by many current developments, such as wireless sensors and actuator (WSAN) and identifiable evidence of radio recurrence (RFID). Kevin Ashton from Auto ID-Center MIT first introduced the possibility of Internet of things.

Due to the widespread accessibility of the Web as Wi-Fi (3G, 4G LTE), the comprehensive detection has become evident from now on. Thus, an entrance was opened to people and customers who will finally be adding to the shrewd urban areas in future. The amount of gadgets associated with this would increase enormously and reach somewhere between 50 and 100 billion between now and 2020. This enormous number of gadgets can detect and make administrations widely available. The data and correspondence systems are constantly integrated in the IoT worldview into our environment. So detect and prepare different true questions and place the data on distant mist. IoT is a key element in the development of lively homes, skilful cities and a brilliant infrastructure for medical

services. IoT is widely recognised on the opportunity to gain customer confidence by providing strong security and defence. IoT security is one of today's hot explorations. Many analysts worldwide use their efforts to resolve various challenges in IoT security. However, because of its heterogeneity, protection IoT is an exceptional examination. These improvements have their very own standard security and safety defects, which appear to be found in IoT settings, as the web of things is a combination of these innumerable inventions. In this document, the IoT architecture for security evaluation on each IoT layer is briefly discussed. We speak about safety hazards in each layer and possible attacks by enemies. We also suggested counter-measures to alleviate the risk of these attacks. IoT's base is exceptionally inclined towards significant security attacks, such as Denial of Service (DoS), replay assaults, centre station guy, cloning stuff, snoops and steering attacks. For eg, gadget hardening, security failure, data revelation, DoS and spoofing, signal infusion and side channel attack have been explicitly assembled by Atamli et al. Atamli. IoT devices are compulsory and existing cryptographic safety arrangements cannot be used for these gadgets, making them inclined to reliability and confidentiality of information.

Similarly, three Security Goals such as confidentiality, rectitude and accessibility are difficult to achieve with composition work assaults. This paper analyses subtleties on the problems of traditional safety structures in IoT. Additionally a major IoT protection problem is the validation and access control mechanism of gadgets. The huge number of gadgets and machine to machine (M2M) correspondence existence of IoT are the reason for the verification and access control problems in IoT. A double reading of medical pictures can improve the detection of the tumour.

1.1 METHOD OF APPROACHING

Many new conventions are normalised every year with the irregular flow of examination interest for IoT. Overview papers are therefore continually written to feature different IoT-identified aspects of normalisation. Instances in such papers recall the IETF instructions, safety conventions and layer principles as well as applications or transports. Different articles discuss a specific layer of standards,

such as correspondence agreements or guidance. Summarize in particular the core norms provided by different organisations of guidelines until 2015.

It also discusses various IoT problems such as flexibility and adaptability. In this paper, we hope to provide a detailed analysis of new concepts, draughts and conventions which broaden the work that has been done. This allows us to explore further principles, introduce a section of the new standards in the IETF and discuss the cutting-edge agreements on which normalisation will soon take place.

As illustrated in, the IoT environment of the ecosystem includes a seven-layer model: sector, secured, interconnected, incorporated, examined, applied and administrative. The layer on the base could be a genius system, a home-related framework, or shrewd wellness, and so on. Sensors and skilled gadgets that can be called the application centre will be included in the next layer.

The form and distribution of the sensors varies according to the ideal applications. Temperature sensors, mugginess sensors, power metres or cameras are instances of such sensors. The third layer consists of an interconnection layer which promotes sensor information correspondence to a server farm or cloud.

In addition to other information sets such as geology, population or monetary information, the information is included. The combined knowledge is also analysed using IA and IT strategies. Coordinated effort and correspondence programming in new application levels can facilitate these enormous distributed applications.

These principles include organisational programming (SDN), engineering administration (SOA) and so on. Last but not least, the top layer includes administration results such as energy management, board well-being, preparation, transportation etc. For each one of these 7 layers based on each other, security and the board are essential and thus they appear as a background.

The recent study showed how supervised techniques such as the Convolution Neural Network allow for the classification of human brain in magnetic resonance (MR) images. The approach proposed would dramatically reduce the

time needed to diagnose and classify the brain tumour. Increasing the detection efficiency and the entire system dramatically reduces tumour detection time and expense. The main objective of the device proposed is that it can be used by anyone due to its decreased cost for the detection process.

CHAPTER 2

LITERATURE SURVEY

Kamil Samara has suggested energy efficiency, load control and a wireless network routing protocol. The led diffusion algorithm saves power by hop-transference of data packets and flood prevention routes. The goal is not to find the best or the most healthy routes in the directed diffusion algorithm (healthier paths are paths that utilise less total energy than others and avoid critical nodes). The quality of energy use can be increased by the use of healthier roads. He suggested an ideal WSN routing protocol based on the total energy available on the route, running time and avoiding critical nodes to ensure that the safe routes are respected. This decision is taken with information about possible routes, and with non-incremental computer planning, we introduce paths that connect our parameters. Small networks did not show a substantial difference between the selected protocol and the specified algorithm for the simulation performance. The performance of the project protocol has increased as the network size increases.

Cumar Pradeep ATREY² implemented reliable, safe data collection in Puneth and Muralidhar Kulakarni¹'s wireless sensor networks. By randomization and dispersal routing, the feature compensates for H-SPREAD deficiencies. This method enhanced security with unequal dispersion compared to H-SPREAD. The network lifecycle and security are all taken into consideration for an unpredictable and deterministic combination. As a simple implementation process, i.e. nodes (circular) and BS, the approach was very specific (center of the network area). The research looks at the SSR architecture, which allows for secure, energy efficient wireless networks for group communications. This method increases energy efficiency, but does not manage the assault by CN. They have high security and cn attack is not feasible. They have high protection. This process is taking place.

As an alternative to the Wireless sensor networks with handheld dishes, Jesse Whitehead suggested cluster-based confidence proliferation and energising data collection. When it is not possible to find vital and engaged core stations such as tornado or volcanic eruption, mobile sinks should be given to capture data. In order to link nodes securely on the basis of confidence and network clusters, Jesse Whitehead suggested a smooth clustered structure. Jesse Whitehead, the emulation display, shows that the energy conservation and networking life of this project exceeds anyone.

Wireless network sensor management had been implemented by Ahmed Khana, Hassaan Khaliq Qureshia and Adnan Iqbalb Energy. Wired batteries are also used to power the nodes. Any battery powered network service energy saving devices are implemented. The limitation linked to the low power of the battery forced the research trend to find new sources through environmental collection. This study determines a high degree of classification in WSNs for energy efficiency. They list a number of schemes focused on battery power and energy collection schemes. The most recent change to the wireless power transmission in the sensor node is also illuminated as an alternative to conventional batteries. Alternatively, they proposed the conventional energy savings that the sensor network would make when designing energy efficient systems, taking account of recent energy allotments.

In order to collect positive and enthusiastic data on unused wireless networks, Farah Kandah, Jesse Whitehead and Peyton Ball launched it. It is important that important degradation of the system caused by incorrect or inaccurate transfer of data be ignored. Building trust - assurance between two devices guarantees the confidence and accuracy of data, providing a safe mechanism to identify all devices confidently and authorising no particulars in uncertified user details. The output showed that the impact of malicious nodes on the network must be isolated or reduced as the number of loops decreases and therefore decreases energy consumption in the network and increases its lifespan. The benefit of clustering is also the distribution of energy demand across nodes, which prevents node drop and network difficulties at an early stage.

Guosheng Huang has generated a number of energy-efficient data for unmanned aircraft. The maximised trajectory algorithm ant-colony (UAV), which forms the fewest possible data anchors in the field of operation, is given. Secondly, trust and process assessment schemes will be used to identify the trust rating of sensor nodes and only credible data to certify the quality of data collection will be collected. Finally we will find an effective data collection process that would contribute to achieving a much better network power consumption in our TEEDC-UAV scheme, which has been tested on a wide range of experimental output. The network life has increased by 48.9 percent relative to previous approaches. Meanwhile, the trust feature in this article will improve the established precision of a degree of node trust that affects 91 percent of the life of the network.

For the wireless sensor networks, Khalid Haseeb created a heuristic protocol. The proposed protocol allows for heuristic surveys on artificial intelligence to resolve the need for a system for dependence and smart learning. Secondly, transmissions with the least complexity compared with opponent classes are protected for protection. Traffic exploration can help to reduce the loss and network disconnectivity of the pathways management approach. Simulation results indicate that the SEHR protocol increases the efficiency of the network performance by average of 18%, a 42% packet reduction rate, at 26%, a 36% energy consumption of faulty routes, and a 43% numerical rate compared to current work.

Naveed Islam implemented the Lightweight Safe and Efficient Sensor Network Routing Protocol. This work enhances the energy efficiency, utilises the Service Quality (QoS) factor and makes time sensitive applications at network edges more accessible. This protocol is centralised on two primitive layers of encryption, ensuring data in real-time. Cluster heads and fog nodes have a lightweight data confidentiality system first, and an asymmetric high performance framework is introduced to define the layers of cloud and fog. In terms of routing, reliability and network control, the survey of simulation-based research has found important results in relation to current technologies.

Liangrui Tang, Zhilin Lu and Bing Fan suggested that the wireless sensor network routing algorithm be energy efficient and reliable. DS-EERA creates three

guides as residual electricity, traffic, and so forth. They gain entropy weight to monitor the weight of three guides objectively. The DS proof theory fusion rule registers to fuse each index value with the BPA function to pick a next hop after the intent of the basic probability assigning (BPA). Each network node transfers data via this routing approach. DS-EERA is a very promising theoretical paper and simulation result, enabling it to extend its network life effectively. In the meantime, the loss rate of packets can also be reduced and data transfer reliability increased.

Muhammad Ilyas suggested an energy-efficient protocol for Internet routing for trust-based sensor networks. They also implemented a three-tier clustering system with an integrated authentication function to recognise and blacklist malicious sensor nodes. Clustering protocol with the sinknote used for the cluster head and grid head on the same priority. For testing link performance and enhancing routing reliability, hardware-based link level estimators are used. In the end of the day, the effectiveness of the introduced technique was assessed by unnecessary experiments. It exceeds many related protocols such as the futile logic of the hybrid ants-based clusters and colony-based optimization.

A survey on data collection and targeted methods of energy-efficient routing in WSN has been introduced by U.Nilabar Nisha, S.Velmurugan, L.Raja, G.Shanthi. However, energy usage did not decrease and network lifetime was not improved in working methods during the processing, data collection and goal surveillance. Our main goal is to reduce energy consumption and the reliability of data collection in WSN data collection and monitoring. In this study, a survey is carried out using methods to achieve greater energy efficiency and maximum efficiency in data collection with a minimum target monitoring time in WSN.

Venkateswara Rao M has implemented a Zone based Wireless Sensor Network Safe Energy Efficient Attack Resilient Process. This work provides a secure, energy-efficient, and aggressive routing solution for zone-based wireless sensor networks including proactive monitoring and malicious attack prevention. This study divides the network into areas and gives the profit rate of the spreading packet across the region, a probabilistic and fused score that differs from previous studies in the observation of any malignant node. The routing can be

changed to accommodate continuing treatments for energy and protection. A strong evaluation that can influence the network operations cannot determine the interaction of the systems over the life cycle of the network. In NS2, the solution applied is checked and work alternatives compared to demonstrate the effectiveness of the method.

2.1 COMPARISON OF LITERATURE SURVEY

AUTHOR	PROBLEM ADDRESSED	METHODOLOGY PROPOSED	REMARKS
Kamil Samara (2020)	Routing Algorithm	Direct diffusion algorithm	Simulation result did not point out a significant difference in smaller networks
Deepa Puneeth, Nishanth Joshi, Pradeedp Kumar Atrey, Muralidhar Kulakarni (2020)	Security	H-Spread	In this method we can achieve low security
Jesse Whitehead (2020)	Time Management	Cluster based mechanism	Inference high performance
Junaid Ahmed Khana, Hassaan khaliq qureshine, Adan Iqbalb (2019)	Energy Management	Battery driven energy consumption	High performance
Farah Kandah ¹ , Jesse Whitehead ² , Peyton Ball.(2019)	Energy consumption	Reducing number of nodes	High performance

CHAPTER 3

AIM AND SCOPE

It mainly uses the remote sensor network based on the occasion divulgation and data range. The hubs often communicate information to the base station via multi-jumps due to their restricted correspondence gap between the sensor hubs. Instructions for finding a way to guide calculations with lowest energy use from a source hub to a sink hub. Figure 1 shows the geography of WSN. By multi-bounce sending of the collected data, the data assortment terminals are viewed as the fundamental hub.

Assume there are several regular hubs and a lonely sink hub in the centre in a sensor network with a territory $L = L$, and not all hubs are, shifted by sending. The zone between the hubs is fixed. Both sensor hub systems are isomorphic, have capacities to steer, send and accept, two hub can be transmitted in single or multi-bounce mode and the subordinate state is very similar.

The heap of hubs underneath is 0, and energy underlying is E_0 . The hubs trading data approach is the same. Each hub has an interesting ID, supporting the storage of data such as remaining power, package ID, next bounce ID, sender ID and so on. These data are constantly refreshed as the next person changes. It is noted that hubs have been limited in traffic lengths and that a first-in first-out (FIFO) package of information is prepared.

3.1 EXISTING SYSTEM

In view of its easy scaling and multisectoral characteristics, the promotion of competent remote communications and the advance of electronic data innovation (WSN) is widely used in various fields. Nonetheless, the WSN hubs are typically battery powered and generally shipped outside or under harmful conditions, making

them poorly equipped for energy regeneration. The cost of excess shipment and replacement of hub is typically high. A efficient steering procedure is therefore supposed to restrict the use of the network's resources and take lifetime. Since the power of the sensor hub is mostly used for the collection and transfer of information, the traditional directions approach mainly thinks how to use the restricted means of moving information from the source hub to the target as quickly as possible. In any case, a lot of information is transmitted via a 'Many-to-one mode' from the source hub in the energy-induced sensor organisation, with genuine 'Channel Effects' and 'Energy Opening' concerns being effective.

3.2 PROPOSED SYSTEM

The "many-to-one" mode of communication can also cause the organisation to block its data. For instance, when a main occasion is set, the hubs must send a lot of information in a short time to the sink hub. Clog will dispose of a huge number of information packs, reducing the unequivocal efficiency of data transfer. In the meantime the organisation, because of data retransmissions that reduce energy consumption, would produce superfluous energy utilisation and an increased transmission time. Some blockage scans, when blocking is discovered, usually use such acts to monitor the inflow speed of information, and it is difficult for the upstream hub to manage the rate of information flowing upstream from blocking. Many steering systems often like hubs to reduce clogs, but because of diversions the hubs can lose more energy. Therefore we propose an energy efficient and solid guiding calculation based upon ECC, by receiving multi-characteristic dynamic technique, to enhance efficiency of energy and to increase organisational lifetime and reliability of information transmission. This paper has the main commitments as follows:

Proposed Routing Model constructs the centre evaluation work from the perspective of the reduction in energy use. This means that the hubs' scripts are fully considered and disconnected into three lists: the energy transmission proportion, the degree of inaction and the energy thickness factor.

In view of any round of operation in an organisation, we fairly determine the weight of every record depending on the coefficient of differentiation between entropy estimates.

There are not many WSN use of DS proof hypothesis in hub credits while steering. PROPOSED ROUTING MODEL sees three records as evidence and determines if the hub should become the next hop as its goal of recognition. We use DS combination rules creatively to record the hub wire and take the results as the basis for guiding decisions.

The re-launch results show that PROPOSED ROUTING MODEL is able to effectively decrease energy use as well as to drag out network life in energy-driven networks and also to achieve a reduced package disaster rate and increase information transmission trustworthiness.

3.3 NEIGHBOR DISCOVERY BLOCK

The Neighbor Discovery Block is enabled on network bootstrapping or by presenting each new hub. At this point, each hub sends and sends a one-spring message to get to know the next hubs. The message of welcome is accompanied by: {message type = 1 [one byte], sending ID [one byte], MAC [x bytes]} of the plot, if any, by confirming the MAC using the public key of the sender (hub ID) for PBC, or by the symmetrical key for ECC, as specified by the manager on the initialization block. Should the bundle not be correct, the recipient hub will rundown the sender in its neighbours and reply using a "unicast" which replies the bundle back to the sender to confirm its neighbourhood between them. The message of a welcome response has the same setup as the Message Type = 2 welcome message. Likewise all hubs transmit hello keep alive message to their neighbours, every fixed time, set and synchronised by BS. In the Initialization Block the executive sets the time frame, and the time for submitting movement reports is taken several times.

This welcome message is exactly the same as a welcome message with the Message Type = 3, and is used to ensure that there is a hub that keeps the BS refreshed with right data. The endless provision of a welcome, live message clearly reinforces without a response the situation with its neighbours. Note that if a confirmation plot set should occur, there should be labelled and checked a broad range of Hi messages for confidence in the characters of your neighbour.

3.4 BROADCASTING NODES

Another sort of attack is a preliminary one which triggers the organisation by communicating parcels to all its neighbours continuously. In two phases this kind is recognised and separated. Firstly, because the whole of their non-DL neighbours drop this parcel, the effect of the transmission is privately removed. As regards the fast transmission speed of the hub, the BS identifies that the hub will either wait or detach it immediately, depending on the time frame overhead of what it should be.

CHAPTER 4

ALGORITHM AND IMPLEMENTATION

4.1 ALGORITHMS

4.1.1 *ECC Algorithm (Elliptical Curve Cryptography)*

It is important to get basic climate information in military applications because such complex interaction helps information. The methods used to move the issue led to the suggested work helping to move the basic information safer in the multi-faceted WSN design. Encode the data using insignificant key size was done with the elliptic curve encoding part.

ECC Elliptic Curve Cryptography's essential idea is based on three factors

- The primary number to be selected as the largest
- A state of the curve
- Bending Public Point

Private Key: Private number selected

Key to the public: The public spotted Number of private opportunities In this type of context, the calculation of the private clay from the public key should be elliptical logarithmic work. Elliptic curvature depends on the focus that should meet the requirement. $y^2 = x^3 + ax + b$ Points may be marked with another point in the diagram. There are two areas where you can find an additional point together: $B = C + A$ dab. $C = D + A$ dab $A + D = E$

Confidence estimate: The trust assessment of the neighbourhood nodes is presented in this section. The fundamental comprehension of the trust is summed up as:

Confidence is a connection created for a particular action between two entities. One individual relies in particular on the other to take an action. The first entity is referred to in this study as the subject, the second entity is the agent. Therefore, {subject: agent; action} is the notation used for describing trust relationship.

Allow $T\{\text{subject: agent; action}\}$ to indicate the trust value of the relationship of trust {subject: agent; action} and $P\{\text{subject: agent; action}\}$ to indicate the likelihood of the agent to behave in the subject's viewpoint. The theory of information notes that entropy is an insecurity measure; thus, entropy-based trust value as an error! $I\{T\{\text{trust, agent, action}\}\} = H(p)$, for $0.5 \leq p \leq 1$, $H(p) = -\log_2 p$ – reverse-reduce – reverse-release – reverse-reduce: $<(1)$ where, $H(p) = -\log_2 p$ – reverse-reduce the reverse-release: $P = P[\text{support: agent; actions}]$ The confidence is an ongoing true number in this work $[-1, 1]$. The following characteristics are met by this description.

- If $P = 1$, the subject most trusts the agent and the value of the trust is 1
- If $P = 0$, the subject rejects most of the agent and the confidence value is -1
- The subject does not know the handler, the confidential value is 0 if $P = 0.5$ The trust value is generally negative for $0 < P < 0.5$

$$P(n) = \frac{1}{n(N)} \sum_{k=1}^N P(n(n)k) \quad (4.1)$$

This calculates the likelihood of all the trials by the distribution below of the Bernoulli.

$$N K (N K) K P\{n(N)=K\} = p(1-p)^{-(N-K)} K \{n(N) = K\} = p(1-p)^{-(N-K)} \quad (4.2)$$

P = average chance of success of each packet transmission N = Total number of source packets; K = Number of neighbourhood node-transmitted packets

Secure road creation via antnet: if a node (source) wants to set a path to the other node (destination), the source will first try to find several destination routes. The source then attempts to identify the packet-forwarding credibility of the nodes on the routes from its own confidence record or by asking for recommendations. The source finally selects the reliable data transmission path. The source node will update the trust records following transmission on the basis of its track consistency observation. The following steps lead to the discovery of the route by the use of the Antnet algorithm:

- Every source starts several forward agents packets via multi-hop propagation to the destination. The route is randomly chosen based on the existing routing table
- Forward agent packets build a stack that pushes every visited node during travel times, trust values and traffic intensities as shown in the Figure. Some backward agent packets are sent to the source, when the packets get to the destination. The backward agent packets inherit the parameters of the stacks during this period. The parameter is popped and check again as shown in the figure. The packages for returning agents include the confidence values, the intensities of traffic and delays of the routes to the source. The optimum destination route will be selected from source[8-9].

Figure shows the hidden exchange of the source and the destination and the ECC. The total number of points in Elliptic curve, including the point on infinity. G Point on Elliptical curve with order 'n' N The secret keys NA, NB The public clues PA, PB

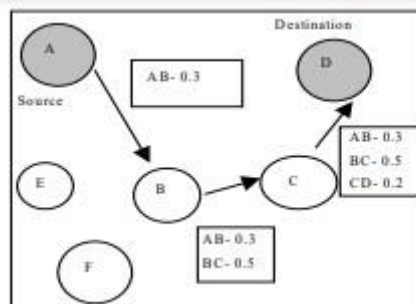


Fig. 4.1 Forward route discovery (ECC)

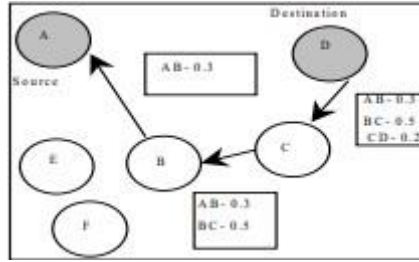


Fig. 4.2 Backward route discovery (ECC)

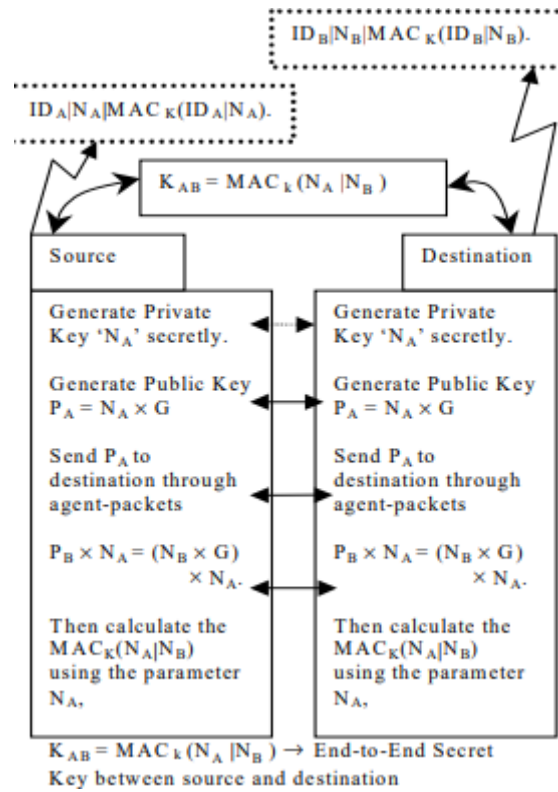


Fig. 4.3 Key generation in ECC

4.1.2 RSA Algorithm (Rivest Shamir Adleman)

RSA Algorithm In this paper, the data is encoded utilizing RSA public key calculation KEY GENERATION FOR RSA ALGORITHM

- It is needed to choose two indivisible numbers p and q haphazardly.
- $n = p * q$ and $\phi(n) = (p-1) * (q-1)$ be registered

- One discretionary whole number „e' is to be chosen with $1 < e < \phi(n)$ to such an extent that $\gcd(e, \phi(n))=1$
- The whole number d is to be registered which fulfills $1 < d < \phi(n)$ and $ed \equiv 1 \pmod{\phi(n)}$
- Return (n,e,d) Public key parts are e and n where as Private key is d. The data is enciphered by utilizing public key of the recipient and translated by utilizing private key of the collector. RSA ENCRYPTION SCHEME Input: The plain content $m \in [0, n - 1]$ and The public key i.e (n, e) Output: Ciphared text c.

$c = m^e \pmod{n}$ must be processed

The encoded text, c must be returned.

RSA DECRYPTION SCHEME Input:

RSA private key d and the encoded text c Output: Plain content m.

$m = c^d \pmod{n}$ must be registered.

The unraveled text m must be returned which is the first content.

We evaluated the time a hub spends well above that needed for ordinary AODV for each overhead test. All experiments on the Zauruses were carried out with only the critical programming (i.e., no graphical login supervisor, no X worker, and so forth) In the SAODV tests, we calculate the age and SSE approval which needs an advanced mark/check and hash calculation. These tests have been conducted in MD5 and a 256-cycle RSA key pair has been used to carry out computerised signature and verification. For every estimate, there were 1000 tasks and 1000 estimates in general. The aftereffects of our SAODV experiments are shown in Table 1. In this way, the hub burns 30.8 milliseconds generating the SSA to send an RREQ, RREP, or HELLO message. The crucial impact on execution is when HELLO messages SSE are generated as they are often sent. In order to agree with AODV in particular, every HELLO INTERVAL millisecond should be sent a HELLO message, unless any messages have been exchanged during the past. This means that single RREQ and RERR messages will prevent the transmission of a

HELLO message, as every other message is unicast. This may, of course, put enormous weight on each hub.

Each RREQ and RREP receives a delay of 3.8 milliseconds at a jump that progresses SAODV because any message with an SSE is accepted prior to any further preparation. HELLO messages also set aside a related effort for approval. Although $\text{ALLOWED HELLO LOSS} \times \text{HELLO INTERVAL}$ milliseconds should pass by until a link is decided and a neighbour should be removed from his director's table, the current status of those neighbours, whose HELLO parcels are deferred in consent, might be hesitant on a hub with a few neighbours and a large number of information to advance. We test the RREP-, HELLO- and R ACK messages by bundle overhead in TAODV. The broad system limits examined in [15] do not affect the TAODV overhead in any of the tests we have conducted. However, these qualities were necessary to render the RSV count. The following system has used wide-ranging considerations for all of the TAODV tests: $l = 0.8$, $p = 0.6$, $ph = 0.4$, $pc = 0.2$, $\alpha_1 = 0.4$, $\alpha_2 = 0.4$, and $\alpha_3 = 0.2$. The small task season saw an estimated 1,000,000 work carried out and an estimate of 5,000 was made. The results for the TAODV tests are shown in Table 2. As the results show, TAODV's overhead per plot is substantially less when compared to SAODV. The R ACK bundles are elaborated by the fundamental source of the overhead. As the R ACK bundles are new parcels rather than parcel increases, it was necessary to give a bundle of cradles each time a R ACK bundle was sent in the message, sending our implementation arrangement. The parcel cushion was designated at that point and the expansion was only composed into open space toward the end with various messages extended. This distinction led to the overhead of the R ACK communication of 0.193ms

4.2 IMPLEMENTATION

NS (version 2) is a tool for re-enacting the network open-source. It is a subject structured in C++ and Otol and is a discrete event based test system. NS is used in network studies for refocusing different forms of wired/distant areas and deep regional associations, network shows such as TCP and UDP, FTP, Telnet, CBR and VBR traffic sources, switch board frames such as Drop Tail, RED and

CQB, and monitoring computations like Dijkstra. In C++ and Otcl, Ns2 is written to detach the use of control and data. The test system supports a leadership class hierarchy in C++ and looks at movement within the Otcl interpreter (deciphered progression).

The inspiration for the use of two languages in ns2 is that various tasks vary. For instance, the restoration of the shows requires profitable bytes and group headers to monitor the runtime. Of course, in-network questioning where a few constraints are contrasted and an opportunity to change the model and run it again is more important in order to rapidly evaluate various circumstances.

In ns2, C++ is used to unmistakably use shows and when everything is said in order to achieve this, when every bundle of a stream has to be ready. For example, C++ is the language of choice, if you need to perform another coating discipline. Naturally, once the strategy and the plan are responsive. Otcl works slowly, but in general the progression of proliferations can be modified rapidly. Ns2 allows complete C++ items to be opened to the Otcl arbiter. Therefore, items from the Tcl level can be regulated from the moment C++.

4.2.1 OTCL Essentials

This section shows the phrases in the Otcl language and some basic orders used by ns2. Before moving to the part that manages to shape the real recovery, it is important to check how Otcl operates.

4.2.2 Assigning Esteems to Factors

In tcl, qualities can be reduced to factors and these qualities can be included in orders in addition:

Fix a set of 5 b [expr \$a/5] [Expr \$a/5]

The vector "5" is designated in the primary line. In the subsequent line, an incentive for variable b is used for the after-effect from the order [expr \$a/5],

which is raised to 1. (including 1). The "\$" symbol is used to obtain a value found in an order substitution variable and square parts are a sign.

4.2.3 *Procedures*

New systems can be characterised by procurement. The primary argument to be procured is the name of the tactic, and the following argument includes an overview of the names of the dispute. For example, the following can be characterised as a method which calculates two numbers:

```
Full proc {a b}
```

```
$a + $b Expr
```

```
}
```

The device below shows a number factory:

```
Factorial provision a {
```

```
{{ $a <= 1 } { on the off chance
```

```
Please return 1
```

```
}
```

```
#here again the device is named
```

```
[factorial [expr$x-1]] Express $x * [factorial]
```

```
}
```

It can also be conceived as a contentious list for an unfulfilled string. Nonetheless, the factors used by the strategy should be defined globally for this situation. Examples include:

```
Proc whole {}
```

```
A b all over the world
```

```
$a + $b Expr
```

```
}
```

4.2.4 *Files and Records*

For the order reading in tcl, a document can be opened:

Set the [open test.dat r] testfile

A overview with an order can be placed on the main line of the document:

Get the list of \$prooffile

It is currently possible for orders to be issued for the components of the rundown (component numbers start from 0):

First placed [lindex \$list 0] Set first

Set second [lindex \$list 1] Set second

A record may essentially be written with an order of suppression:

Sets the [open test.dat w] testfile

Puts "testi" on \$testfile

4.2.5 *Calling Subprocesses*

The Order Manager performs and tightly hangs for completion. Executive use means that the shell programme is supplied with an order line. To delete a record, for example:

```
rm $testfile executive
```

The executive order is particularly important if a tcl material from inside another tclscript has to be called up. For example, to run the example.tcl tcl-content on different occasions and to evaluate the "test" boundary from 1 to 10, you can type the accompanying tcl-contents in the following line:

```
{set ind 1} {$ind <= 10} {including ind} } {  
Put the $ind test  
Executive ns test.tcl  
}
```

4.2.6 *Creating the Geography*

To have the option to run a reproduction situation, an organization geography should initially be made. In ns2, the geography comprises of an assortment of hubs and connections.

Before the geography can be set up, another test system object should be made toward the start of the content with the order:

```
set ns [new Simulator]
```

The test system object has part works that empower making the hubs and the connections, interfacing specialists and so forth All these fundamental capacities can be found from the class Simulator. When utilizing capacities having a place with this class, the order starts with "\$ns", since ns was characterized to be a handle to the Simulator object.

4.2.7 *Nodes*

New hub articles can be made with the order:

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

```
set n3 [$ns node]
```

The part capacity of the Simulator class, called "hub" makes four hubs and appoints them to the handles n0, n1, n2 and n3. These handles can later be utilized when alluding to the hubs. In the event that the hub isn't a switch however an end framework, traffic specialists (TCP, UDP and so on) and traffic sources (FTP,CBR and so forth) should be set up, i.e, sources should be connected to the specialists and the specialists to the hubs, individually.

4.2.8 *Agents, applications and traffic sources*

The most widely recognized specialists utilized in ns2 are UDP and TCP specialists. If there should be an occurrence of a TCP specialist, a few kinds are accessible. The most widely recognized specialist types are:

- Agent/TCP – a Tahoe TCP sender
- Agent/TCP/Reno – a Reno TCP sender
- Agent/TCP/Sack1 – TCP with specific affirmation

The most well-known applications and traffic sources gave by ns2 are:

- Application/FTP – produces mass information that TCP will send
- Application/Traffic/CBR – produces bundles with a steady piece rate
- Application/Traffic/Exponential – during off-periods, no traffic is sent. During on-periods, bundles are produced with a consistent rate. The length of both on and off-periods is dramatically circulated.
- Application/Traffic/Trace – Traffic is produced from a follow record, where the sizes and interarrival seasons of the parcels are characterized.

Notwithstanding these instant applications, it is conceivable to produce traffic by utilizing the techniques given by the class Agent. For instance, on the off chance that one needs to send information over UDP, the technique send(int n bytes) can be utilized at the tcl-level gave that the udp-specialist is first designed and appended to some hub.

The following is a finished illustration of how to make a CBR traffic source utilizing UDP as transport convention and connect it to hub n0:

```
set udp0 [new Agent/UDP]
$ns connect specialist $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 append specialist $udp0
```

```
$cbr0 set packet_size_ 1000
```

```
$udp0 set packet_size_ 1000
```

```
$cbr0 set rate_ 1000000
```

A FTP application utilizing TCP as a vehicle convention can be made and appended to hub n1 similarly:

```
set tcp1 [new Agent/TCP]
```

```
$ns append specialist $n1 $tcp1
```

```
set ftp1 [new Application/FTP]
```

```
$ftp1 append specialist $tcp1
```

```
$tcp1 set packet_size_ 1000
```

The UDP and TCP classes are both kid classes of the class Agent. With the articulations [new Agent/TCP] and [new Agent/UDP] the properties of these classes can be consolidated to the new items udp0 and tcp1. These items are then joined to hubs n0 and n1. Then, the application is characterized and appended to the vehicle convention. At long last, the design boundaries of the traffic source are set. In the event of CBR, the traffic can be characterized by boundaries rate_ (or proportionally interval_, deciding the interarrival season of the parcels), packetSize_ and random_. With the random_ boundary it is conceivable to add some irregularity in the interarrival seasons of the bundles. The default esteem is 0, implying that no irregularity is added.

4.2.9 Traffic Sinks

In the event that the data streams are to be ended without handling, the UDP and TCP sources must be associated with traffic sinks. A TCP sink is characterized in the class Agent/TCP Sink and a UDP sink is characterized in the class Agent/Null.

A UDP sink can be appended to n2 and associated with udp0 in the accompanying manner:

```
set invalid [new Agent/Null]
```

```
$ns join specialist $n2 $null
```

```
$ns associate $udp0 $null
```

A standard TCP sink that makes one affirmation for every a got parcel can be appended to n3 and associated with tcp1 with the orders:

```
set sink [new Agent/Sink]
```

```
$ns join specialist $n3 $sink
```

```
$ns associate $tcp1 $sink
```

There is likewise a more limited approach to characterize associations between a source and the objective with the order:

```
$ns make association <srctype> <src> <dsttype> <dst> <pktclass>
```

For instance, to make a standard TCP association somewhere in the range of n1 and n3 with a class ID of 1:

```
$ns make association TCP $n1 TCPSink $n3 1
```

One can undoubtedly make a few TCP-associations by utilizing this order inside a for-circle.

4.2.10 *Links*

Connections are needed to finish the geography. In ns2, the yield line of a hub is executed as a component of the connection, so when making joins the client likewise needs to characterize the line type.

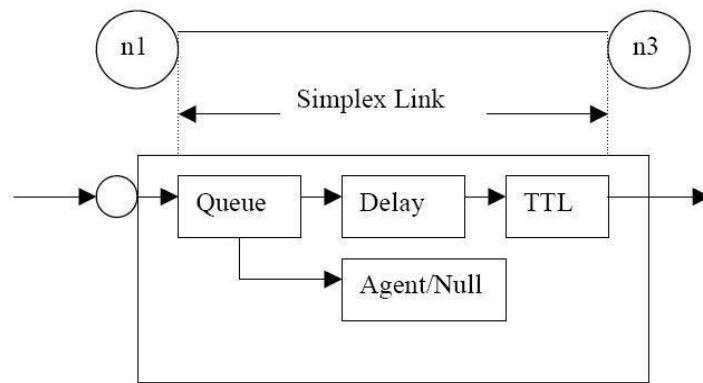


Fig. 4.4 Link in NS2

This Figure shows the development of a simplex connection in ns2. On the off chance that a duplex-interface is made, two simplex connections will be made, one for every course. In the connection, bundle is first enqueued at the line. After this, it is either dropped, passed to the Null Agent and liberated there, or dequeued and passed to the Delay object which mimics the connection delay. At long last, the TTL (time to live) esteem is determined and refreshed.

Connections can be made with the accompanying order:

```
$ns duplex/simplex-connect endpoint1 endpoint2 data transfer capacity
defer line type
```

For instance, to make a duplex-connect with DropTail line the executives somewhere in the range of n0 and n2:

```
$ns duplex-interface $n0 $n2 15Mb 10ms DropTail
```

Making a simplex-interface with RED line the executives somewhere in the range of n1 and n3:

```
$ns simplex-interface $n1 $n3 10Mb 5ms RED
```

The qualities for data transmission can be given as an unadulterated number or by utilizing qualifiers k (kilo), M (super), b (cycle) and B (byte). The postponement can likewise be communicated in a similar way, by utilizing m (milli)

and u (micro) as qualifiers. There are a few line the board calculations actualized in ns2, yet in this activity just Drop Tail and RED will be required.

4.2.11 *Tracing and Checking*

To have the option to compute the outcomes from the reenactments, the information must be gathered by one way or another. NS2 upholds two essential checking abilities: follows and screens. The follows empower recording of parcels at whatever point an occasion, for example, bundle drop or appearance happens in a line or a connection. The screens give a way to gathering amounts, for example, number of parcel drops or number of showed up bundles in the line. The screen can be utilized to gather these amounts for all bundles or only for a predetermined stream (a stream screen).

4.2.12 *Traces*

All occasions from the reenactment can be recorded to a document with the accompanying orders:

```
set trace_all [open all.dat w]
$ns follow all $trace_all
$ns flush-follow
close $trace_all
```

To start with, the yield document is opened and a handle is connected to it. At that point the occasions are recorded to the document determined by the handle. At long last, toward the finish of the recreation the follow support must be flushed and the record must be shut. This is typically finished with a different completion system. On the off chance that connections are made after these orders, extra articles for following (EnqT, DeqT, DrpT and RecvT) will be embedded into them.

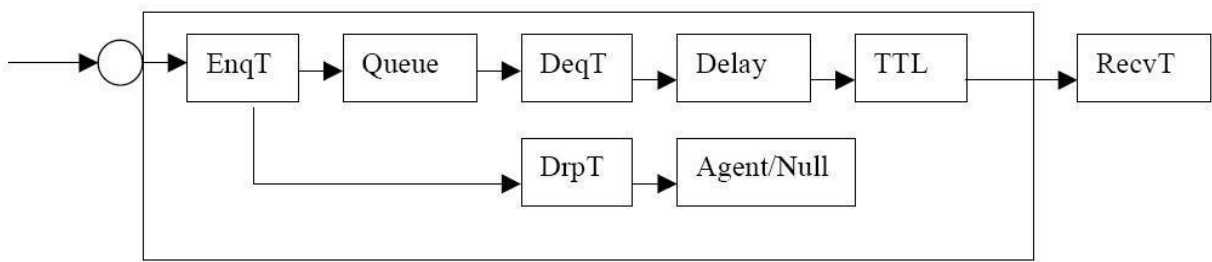


Fig. 4.5 Link in NS2 when tracing is enabled

New items will at that point keep in touch with a follow record at whatever point they get a bundle. The configuration of the follow document is following:

```

+ 1.84375 0 2 cbr 210 - 0 0.0 3.1 225 610
- 1.84375 0 2 cbr 210 - 0 0.0 3.1 225 610
r 1.84471 2 1 cbr 210 - 1 3.0 1.0 195 600
r 1.84566 2 0 ack 40 - 2 3.2 0.1 82 602
+ 1.84566 0 2 tcp 1000 - 2 0.1 3.2 102 611
- 1.84566 0 2 tcp 1000 - 2 0.1 3.2 102 611

+ : enqueue
- : dequeue
d : drop
r : get

```

The fields in the follow document are: kind of the occasion, recreation time when the occasion happened, source and objective hubs, bundle type (convention, activity or traffic source), parcel size, banners, stream id, source and objective locations, grouping number and bundle id. Notwithstanding following all occasions of the reproduction, it is additionally conceivable to make a follow object between a specific source and an objective with the order:

\$ns make follow <type> <file> <src> <dest>

where the sort can be, for example,

- Enque – a parcel appearance (for example at a line)

- Deque – a bundle flight (for example at a line)
- Drop – bundle drop
- Recv – bundle get at the objective

Following all occasions from a reenactment to a particular document and afterward figuring the ideal amounts from this record for example by utilizing perl or awk and Matlab is a simple way and appropriate when the geography is moderately straightforward and the quantity of sources is restricted. In any case, with complex geographies and numerous sources this method of gathering information can turn out to be excessively lethargic. The follow records will likewise burn-through a lot of circle space.

4.2.13 *Monitors*

With a line screen it is conceivable to follow the insights of appearances, flights and drops in either bytes or bundles. Alternatively the line screen can likewise keep a fundamental of the line size over the long haul.

For example, if the hubs n0 and n1 are connected, the line screen can be set as follows:

```
set qmon0 [$ns screen line $n0 $n1]
```

The bundle appearances and byte drops can be followed the orders:

```
set parr [$qmon0 set parrivals_]
```

```
set bdrop [$qmon0 set bdrops_]
```

4.2.14 *Controlling the recreation*

After the geography of reproduction, specialists are created, and so on, recreation begins and ends and various occasions must be reserved. The reconstruction can be started and stopped by order

```
$ns at $simtime "finish"
```

```
$ns run
```

The main order expects to complete the process towards the completion of the reenactment, and then the recreation begins. The completion plan must be designed to flush out follow-up assistance, close the documents and finish off the programme. Alternatively, it will begin NAM, post-cycle data and plot this data. This is also possible.

The completion method needs to contain at any rate the accompanying components:

```
proc finish {} {  
    worldwide ns trace_all  
  
    $ns flush-follow  
  
    close $trace_all  
  
    leave 0  
}
```

Different occasions, for example, the beginning or halting occasions of the customers can be planned for the accompanying way:

```
$ns at 0.0 "cbr0 start"  
  
$ns at 50.0 "ftp1start"  
  
$ns at $simtime "cbr0 stop"  
  
$ns at $simtime "ftp1 stop"
```

If you have characterized your own programs, you may also prepare to start the process regularly, for instance, in the following way:

4.3 SIMPLE STIMULATION EXAMPLE

This section offers a simple NS reproduction and clarifies what each line does. Model 3 is the OTcl material for the basic organizational arrangement and for Figure 4 to reenact. In Example 3 of both of those records "ns-simple.tcl" and "nS-simple.tcl" in the short shell to perform this reconstruction, repeat the code

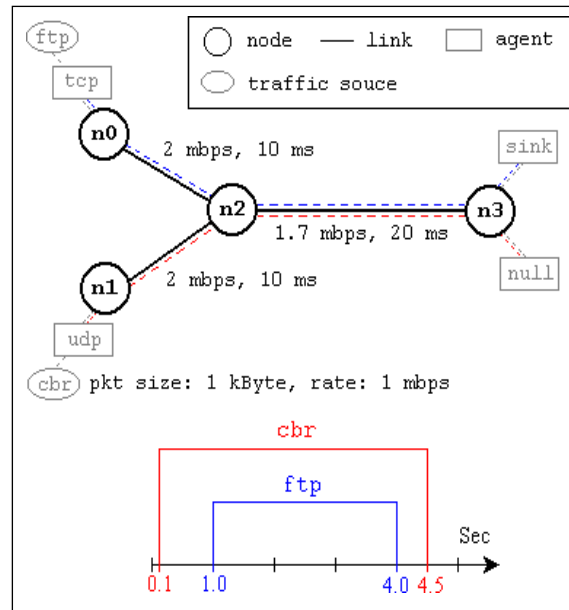


Fig. 4.6 A Simple Network Topology and Stimulation Scenario

There are 4 nodes (n0, n1, n2, n3) on the network as indicated above. There are two bandwidth Mbps and 10 ms of late duplex connections between n0 and n2 and n1 and n2. The bandwidth and 20 ms of delay are 1.7 Mbps of the duplex between n2 and n3. The maximum size for each node is 10 DropTail queues. A "tcp" agent is attached to n0, and the tcp "sink" agent is connected to n3. The parameters that a tcp agent will create are 1KByte by default. A tcp "sink" agent generates the received ACK packets and sends them to the sender (tcp agent) and frees them. A "udp" agent attached to n1 is bound to n3 by a "empty" agent. A 'zero' agent just frees the obtained packets. A traffic generator "ftp" and a "cbr" are connected to the "tcp" and the "udp" agents, and a "cbr" is configured to produce 1 KByte of 1 Mbps packets. The "cbr" starts at 0.1 sec and stops at 4.5 sec and the "ftp" starts at 1.0 sec and stops at 4.0 seconds.

```

#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns nam trace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]

```

```

$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation

```

Example. A Simple NS Simulation Script

Coming up next is the clarification of the content above. By and large, a NS content beginnings with making a Simulator object example.

set ns [new Simulator]: creates a NS test system object occasion, and appoints it to variable ns (*italics is utilized for factors and qualities in this segment*). What this line does is the accompanying:

- Create a scheduler (default is schedule scheduler)

- The "Test system" object has part works that do the accompanying:
- Create compound items like hubs and connections (depicted later)
- Connect network part protests made (ex. join specialist)
- Set network segment boundaries (generally for compound articles)
- Create associations between specialists (ex. make association between a "tcp" and "sink")
- Specify NAM show choices

Etc.

A large portion of part works are for reenactment arrangement and booking, anyway some of them are for the NAM show. The "Test system" object part work usage are situated in the "ns-2/tcl/lib/ns-lib.tcl" document's color fid color: is to set color of the packets for a flow specified by the flow id (fid). This is for the NAM display, and has no effect on the actual simulation.

- \$ns namtrace-all document descriptor: It advises the test system to record reenactment follows in NAM input design. It additionally gives the document name that the follow will be kept in touch with later by the order \$ns flush-follow. Likewise, the part work follow everything is for recording the recreation follow in an overall configuration.
- proc finish {}: is called after this reproduction is more than by the order \$ns at 5.0 "finish". In this capacity, post-reenactment measures are indicated.
- set n0 [\$ns node]: The part work hub makes a hub.
- \$ns duplex interface node 1 node 2 defer line style transmission capacity: providing a couple of easy connections of the speed and delay of the transfer and combining these two hubs. In NS, the yield lines of a hub are performed as a link, so customers can decide the

type of lines during connections. DropTail line is used for the above reproduction material.

- `$ns node1 node2 line limits number`: this line sets the distance between node 1 and Node2 and the number indicated for all single-line connections.
- `Operation node1 $ns duplex-connect`: several lines are used for showing the NAM. This code line is used to screen the link line between node 1 and node 2 hubs. The customer can see the parcels on the line and the customer can eventually also feel the drop.
- The following activities are conducted in accordance with the fundamental organizational structure to coordinate traffic experts such as TCP and UDP, traffic sources such as FTP and CBR as well as to individual hubs and specialists.
- `Set tcp (new Agent/TCP)`: This line becomes a TCP expert. A client can name these elements in order to create traffic specialists or sources (Agent/TCP, Agnet/TCPSink, Application/FTP, etc.).
- `$ns enter specialist platform specialist`: a specialist object built on a hub is linked to a specialist object. This ability in fact refers to the additional part of the indicated center, which links the specialist provided to itself. So, by connecting `$n0 $tcp`, for example, a client may do something very similar.
- `$ns interface agent1 agent2`: The following is to establish a coherent organizational partnership between two experts who can speak with one another.
- Accepting that all the organization design is done, the following activity is compose a recreation situation (for example reenactment planning). The Simulator object has many booking part works.

Notwithstanding, the one that is generally utilized is the accompanying:

- \$ns at time "string": This part capacity of a Simulator object makes the scheduler (scheduler_ is the variable that focuses the scheduler object made by [new Scheduler] order toward the start of the content) to plan the execution of the predefined string at given recreation time. For instance, \$ns at 0.1 "\$cbr start" will settle on the scheduler decision a beginning part capacity of the CBR traffic source object, what begins the CBR to communicate information.
- After all organization setups, booking and post-reenactment system determinations are done, the solitary thing left is to run the recreation. This is finished by \$ns run.

4.4 EXTENDING NS: CREATING ANOTHER AGENT

One can broaden NS by adding new conventions. This segment will examine about how another specialist can be made in NS utilizing a model. The code in this segment executes a type of straightforward 'ping' convention. One hub will send n client characterized number bundles, each in turn at standard stretches, to another hub which will restore the parcel right away. For every bundle the sender will at that point ascertain the full circle time.

4.5 THE HEADER RECORD

First we need to make a header document "myPing.h" in which we will characterize the information structure for the myPing parcel header.

```

struct hdr_myPing {

    char ret;

    double send_time;

    // Header access methods

    static int offset_; // required by PacketHeaderManager

    inline static int& offset() { return offset_; }

    inline static hdr_myPing* access(const Packet* p) {

        return (hdr_myPing*) p->access(offset_);

    }

};

```

When the singe "ret" is on the way from the sender to the ping pit, it is set to '0' and on its way out, the singe "ret" is set to "1." The double time 'sends' is a time stamp determined to be delivered to the bundle. The sender then uses this time to determine the complete circle time. The rest of the software codes in the header are used for any parcel connection to the bundle header.

This code is a subset of the class 'My Ping Agent' as a 'Specialist' underclass.

```

class MyPingAgent : public Agent {

public:

    MyPingAgent();

    int command(int argc, const char*const* argv);

    void recv(Packet*, Handler*);

    void send_packets();

protected:

    int num_packets_;

    double wait_interval_;

    MyPingAgentTimer timer_;

    int count;

};

```

The num parcels vector contains the entire number of packages to be transmitted by the transmitter and the time between two progressive packets is "stand-by-stretch." The 'tally' vector maintains packet inspection to date. We must also characterize a clock controller (My ping agent timer) for the transmitter so that it can reprogram every time the bundle is sent. My Ping Agent uses a My Ping Agent Timer variable 'clock' to track time. The source code for the "myPing.h" header record is given below:

```
/* myPing.h */
#ifndef ns_myPing_h
#define ns_myPing_h
#include "agent.h"
#include "tclcl.h"
#include "packet.h"
#include "address.h"
#include "ip.h"
class MyPingAgent;
struct hdr_myPing {
    char ret;
    double send_time;

    // Header access methods
    static int offset_; // required by PacketHeaderManager
    inline static int& offset() { return offset_; }
    inline static hdr_myPing* access(const Packet* p) {
        return (hdr_myPing*) p->access(offset_);
    }
};
class MyPingAgentTimer : public TimerHandler {
public:
    MyPingAgentTimer(MyPingAgent *t) : TimerHandler() { t_ = t; }

protected:
    virtual void expire(Event *e);
    MyPingAgent *t_;
class MyPingAgent : public Agent {
public:
    MyPingAgent();
    int command(int argc, const char*const* argv);
    void recv(Packet*, Handler*);
    void send_packets();
protected:
    int num_packets_;
    double wait_interval_;
    MyPingAgentTimer timer_;
    int count;
};
#endif
```

4.5.1 *Table of MyPing.h*

The C++ code for new Agent

First the linkage between the C++ code and Tcl code must be characterized.

```
static class kangAbHeaderClass : public PacketHeaderClass {
    public:
        kangAbHeaderClass() : PacketHeaderClass
            ("PacketHeader/MyPing", sizeof(hdr_myPing)) {
            bind_offset(&hdr_myPing::offset_);
        }
} class_kangAbhdrSnd;

static class MyPingAgentClass : public TclClass {
    public:
        MyPingAgentClass() : TclClass("Agent/MyPing") {}
        TclObject* create(int, const char*const*) {
            return (new MyPingAgent());
        }
} class_myPing;
```

The following piece of code is the constructor for the class 'MyPingAgent'. It ties the factors which must be gotten to both in Tcl and C++.

```
MyPingAgent::MyPingAgent() : Agent(PT_MYPING), timer_(this), count(0)
{
    bind("packetSize_", &size_);
    bind("num_packets_", &num_packets_);
    bind("wait_interval_" &wait_interval_);
}
```

When an order Tcl for the class 'My Ping Agent' is executed, the capability 'order()' is taken into account. This will be "sender \$" for our situation (that is expected to be "sender" of the agent/my ping class), since we have to send ping plots from the agent to another ping specialist. In the 'order()' work, we need to parse the order. If the order matches the order, the "send () packets" work will be done, which in turn sends each bundle and reprograms to send the next parcel after the stand. If no match is found, it is transferred into the basic class capability of the 'order()' (the 'agent::command()') for this situation) with its contentions.

```

int MyPingAgent::command(int argc, const char*const* argv)
{
    if (argc == 2) {
        if (strcmp(argv[1], "send") == 0) {
            send_packets();
            // return TCL_OK, so the calling function knows that
            // the command has been processed
            return (TCL_OK);
        }
    }
    return (Agent::command(argc, argv));
}

void MyPingAgent::send_packets()
{
    if(count++ < num_packets_)
    {
        Packet *pkt = allocpkt();
        hdr_myPing *hdr = hdr_myPing::access(pkt);
        // Set the 'ret' field to 0, so the receiving node knows
        // that it has to generate an echo packet
        hdr->ret = 0;
        // Store the current time in the 'send_time' field
        hdr->send_time = Scheduler::instance().clock();
        // Send the packet
        send(pkt, 0);
    }
    if(count < num_packets_)
        timer_.resched(wait_interval_);
}

```

The 'recv()' function determines what measures to take when you receive a packet. If "ret" is 0, the packet has to be returned with the same value for the field "send time," but the field "ret" set to 1. If 'ret' is 1, it is measured and printed for the roundtrip time of the packet. The full "myPing.cc" source code is given as follows:

```

/* myPing.cc */

#include "myPing.h"

int hdr_myPing::offset_;

static class kangAbHeaderClass : public PacketHeaderClass {
public:
    kangAbHeaderClass() : PacketHeaderClass("PacketHeader/MyPing",
                                             sizeof(hdr_myPing)) {
        bind_offset(&hdr_myPing::offset_);
    }
} class_kangAbhdrSnd;

static class MyPingAgentClass : public TclClass {
public:
    MyPingAgentClass() : TclClass("Agent/MyPing") {}
    TclObject* create(int, const char*const*) {
        return (new MyPingAgent());
    }
} class_myPing;

/* constructor*/
MyPingAgent::MyPingAgent() : Agent(PT_MYPING), timer_(this), count(0)
{
    bind("packetSize_", &size_);
    bind("num_packets_", &num_packets_);
    bind("wait_interval_", &wait_interval_);
}

/* if timer expires just do a timeout */
void MyPingAgentTimer::expire(Event*)
{
    t_>send_packets();
}

int MyPingAgent::command(int argc, const char*const* argv)
{
    if (argc == 2) {
        if (strcmp(argv[1], "send") == 0) {
            send_packets();
            // return TCL_OK, so the calling function knows that
            // the command has been processed
            return (TCL_OK);
        }
    }
}

```

```

void MyPingAgent::send_packets()
{
    if(count++ < num_packets_)
    {
        Packet *pkt = allocpkt();
        hdr_myPing *hdr = hdr_myPing::access(pkt);
        // Set the 'ret' field to 0, so the receiving node knows
        // that it has to generate an echo packet
        hdr->ret = 0;
        // Store the current time in the 'send_time' field
        hdr->send_time = Scheduler::instance().clock();
        // Send the packet
        send(pkt, 0);
    }
    if(count < num_packets_)
        timer_resched(wait_interval_);
}

void MyPingAgent::rcv(Packet* pkt, Handler*)
{
    hdr_myPing *hdr=hdr_myPing::access(pkt);
    if (hdr->ret == 0) {
        // Send an 'echo'. First save the old packet's send_time
        double stime = hdr->send_time;
        // Discard the packet
        Packet::free(pkt);
        // Create a new packet
        Packet* pktret = allocpkt();
        // Access the Ping header for the new packet:
        hdr_myPing* hdrret = hdr_myPing::access(pktret);
        // Set the 'ret' field to 1
        hdrret->ret = 1;
        // Set the send_time field to the correct value
        hdrret->send_time = stime;
        // Send the packet
        send(pktret, 0);
    }
    else {
        // Get the receiving time of the packet
        double rcv_time = Scheduler::instance().clock();
        printf("RTT = %lf\n",rcv_time-hdr->send_time);
        // Discard the packet
        Packet::free(pkt);
    }
}

```

4.5.2 Table of myPing.cc

Presently save these two records "myPing.h" and "myPing.cc" in the "... ./ns-allinone-2.31/ns-2.31/" envelope. For great practice make a different index for all projects for client characterized conventions. In our model we have kept these two records in "... ./ns-allinone-2.31/ns-2.31/bwtools/MyPing" registry.

4.6 NECESSARY CHANGES IN NS

Subsequent stage is to adjust a portion of the NS framework records so the recently composed convention or specialist can work with the test system. The records that are should have been adjusted are appeared inside red air pockets

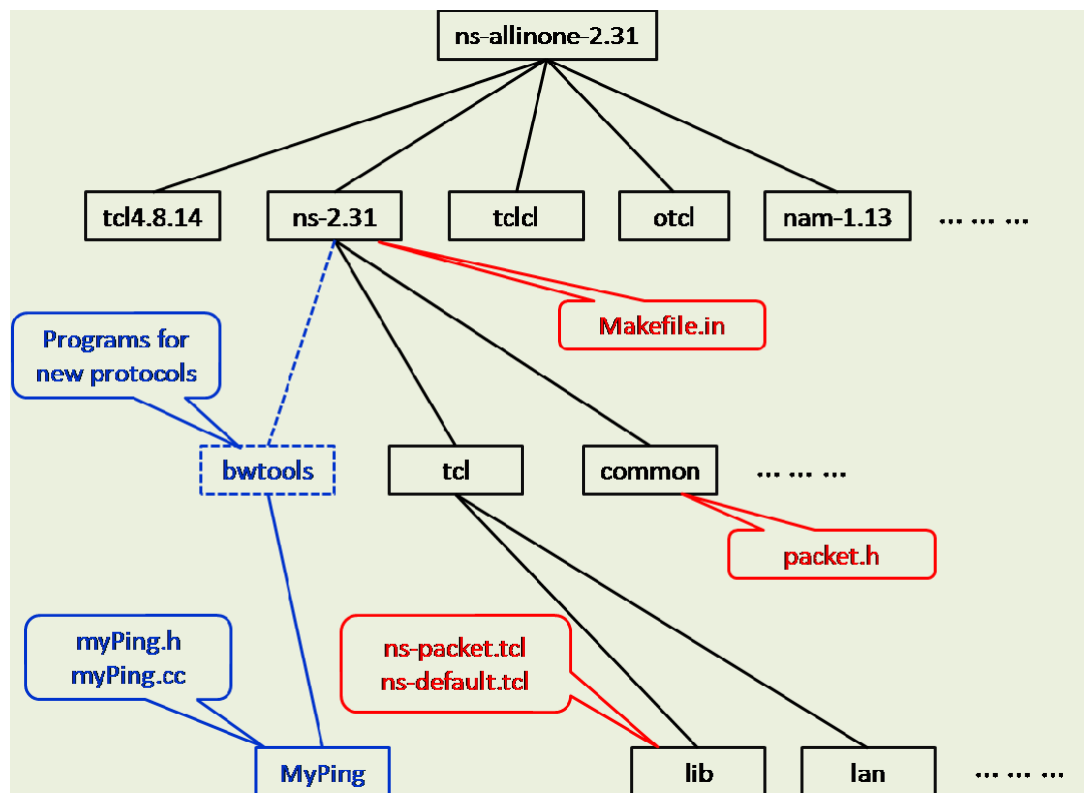


Fig. 4.7 NS directory structure

CHAPTER 5

RESULTS AND DISCUSSION

5.1 EDIT PACKET.H FILE

We need another parcel type for our MyPing specialist, in this way the initial step is to adjust the "\$NS/normal/packet.h" record (expecting that the \$NS climate variable focuses the/ns-allinone-2.31/ns-2.31 area). There you can discover the definitions for the bundle convention IDs (for example PT_TCP, PT_TELNET, and so forth) Add another definition for PT_PING there. In my altered variant of packet.h, the last couple of lines of enum packet_t{} resembles the accompanying code (it may look somewhat changed in changed NS discharges).

```
enum packet_t {
    PT_TCP,
    PT_UDP,
    .....
    // insert new packet types here
    PT_TFRC,
    PT_TFRC_ACK,
    PT_MYPING, // packet protocol ID for MyPing agent
    PT_NTTYPE // This MUST be the LAST one
};
```

You also have to edit the p_info() in the same file to include "MyPing".

```

class p_info {
public:
    p_info() {
        name_[PT_TCP]= "tcp";
        name_[PT_UDP]= "udp";
        .....
        name_[PT_TFRC]= "tcpFriend";
        name_[PT_TFRC_ACK]= "tcpFriendCtl";

        name_[PT_MYPING]="MyPing";

        name_[PT_NTTYPE]= "undefined";
    }
    .....
}

```

5.2 EDIT NS-PACKET.TCL DOCUMENT

Presently in "\$NS/tcl/lib/ns-packet.tcl" record look for the part for each prot { .. }. Inside this part add a passage for MyPing. It will look like after:

```

foreach prot {
# Common:
    Common
    Flags
    IP      # IP
    ... ..
    Pushback      # Pushback, router-to-router
    MyPing
# Multicast:
    aSRM  # mcast/srm.cc
    ... ..
# Other:
    Encap  # common/encap.cc
    IPinIP  # IP encapsulation
    HDLC    # High Level Data Link Control
}{
    add-packet-header $prot
}

```

```

foreach prot {
# Common:
    Common
    Flags
    IP      # IP
    ... ..
    Pushback      # Pushback, router-to-router
    MyPing
# Multicast:
    aSRM  # mcast/srm.cc
    ... ..
# Other:
    Encap  # common/encap.cc
    IPinIP  # IP encapsulation
    HDLC    # High Level Data Link Control
}{
    add-packet-header $prot
}

```

5.3 EDIT NS-DEFAULT.TCL

The default file for Tcl objects "\$NS/tcl/lib/ns-default.tcl" is the default values for Tcl objects. Please insert the next row to specify packet size, packet number and Agent/MyPing waiting interval.

```

Agent/MyPing set packetSize_ 200
Agent/MyPing set num_packets_ 3
Agent/MyPing set wait_interval_ .001

```

5.4 EDIT MAKEFILE.IN

The most recent change is the modification of "\$NS/Makefile.in." You must add "myPing.o" to the object file lists for ns. My version shows the final lines of the edited list:

```

OBJ_CC = \
    tools/random.o tools/rng.o tools/ranvar.o common/misc.o common/timer-
handler.o \
    common/scheduler.o common/object.o common/packet.o \
    ... ..
    bwtools/MyPing/myPing.o \
    tools/trafgen.o trace/traffictrace.o tools/pareto.o \

```

5.5 RECOMPILE NS

We must recompile the NS simulator after changes are made to the NS files. Use the following command for this purpose:

```
droy@aggarp3:~$ cd $NS
droy@aggarp3:ns-2.31$ pwd
/usr/share/ns-allinone-2.31/ns-2.31
droy@aggarp3:ns-2.31$ sudo ./configure
droy@aggarp3:ns-2.31$ sudo make
```

5.6 Simple example using MyPing agent

After recompiling the NS simulator we can use our MyPing agent in our simulations. Following code is simple tcl program that uses the MyPing agent.

```
set ns [new Simulator]
set nf [open out.tr w]
$ns trace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the trace file
    close $nf
    #Execute nam on the trace file
    exit 0
}
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 10Mb 50ms DropTail

#####Sender Side Stuff#####
set sender [new Agent/MyPing]
$ns attach-agent $n0 $sender
$sender set packetSize_ 1500
$sender set num_packets_ 5
$sender set wait_interval_ .05

#####Receiver Side Stuff
set recv [new Agent/MyPing]
$ns attach-agent $n1 $recv

#####Connect the sender with receiver
$ns connect $sender $recv

#Schedule events for the CBR agent
$ns at 0.0 "$sender send"
$ns at 10.0 "finish"
$ns run
```

5.7 RSA – NAM RESULTS

Algorithm : RSA (Rivest Shamir Adleman)
Input data : hello-2021
Transmitter node : 5
Receiver node : 0
Total nodes : 41

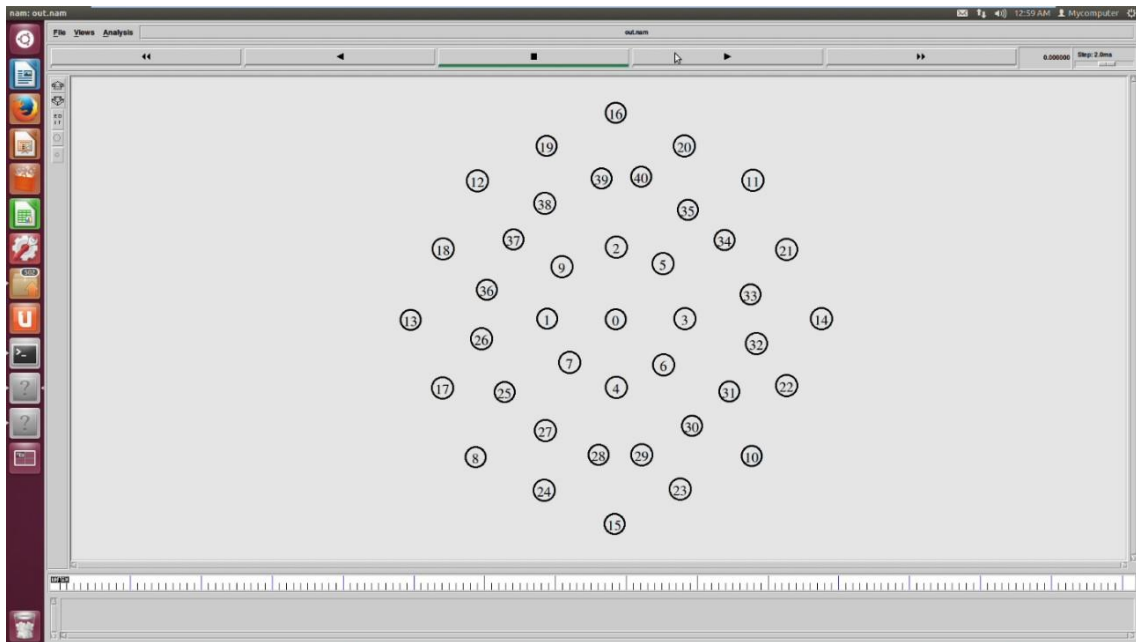


Fig. 5.1 Network Representation NAM output file (RSA)

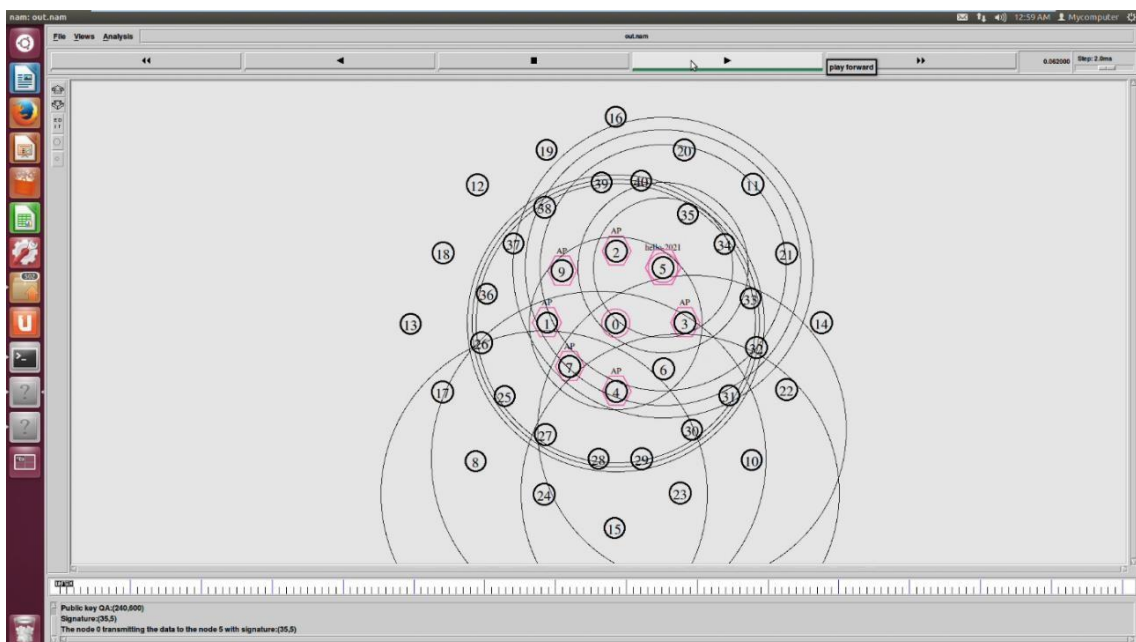


Fig. 5.2 Authentication between Nodes

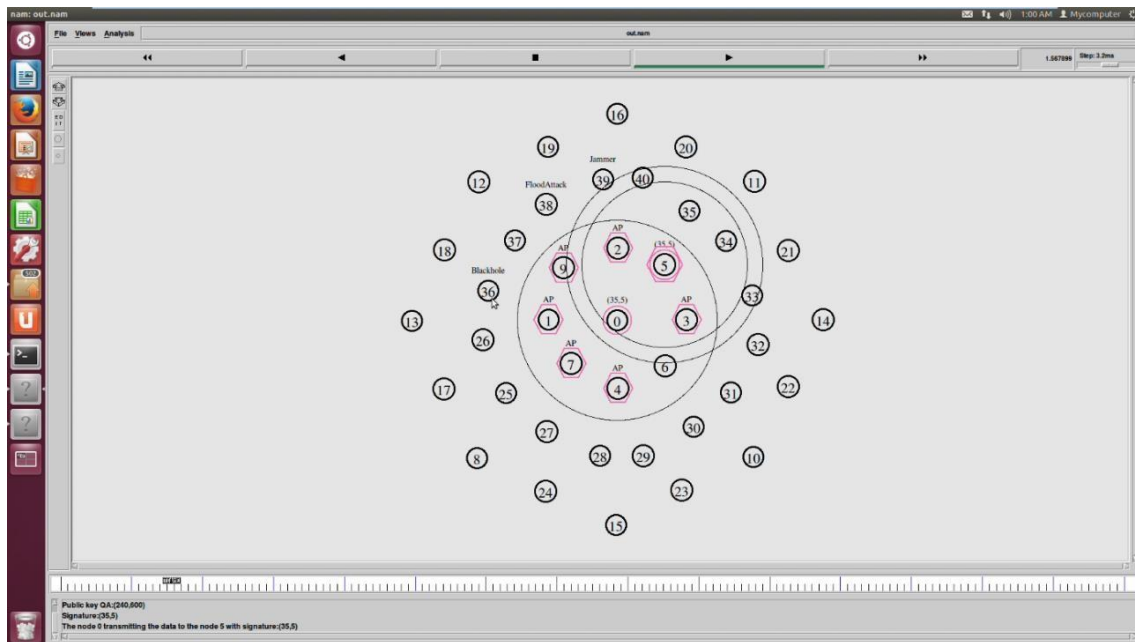


Fig. 5.3 Validating signature (keys) in Network

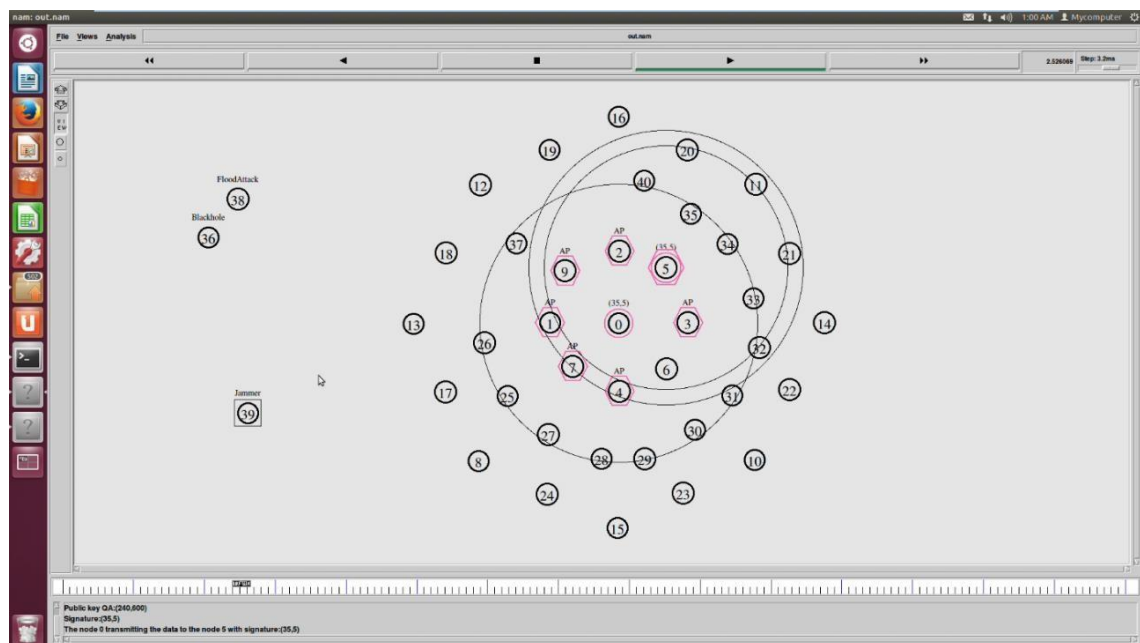


Fig. 5.4 Routing in Network over off the attackers

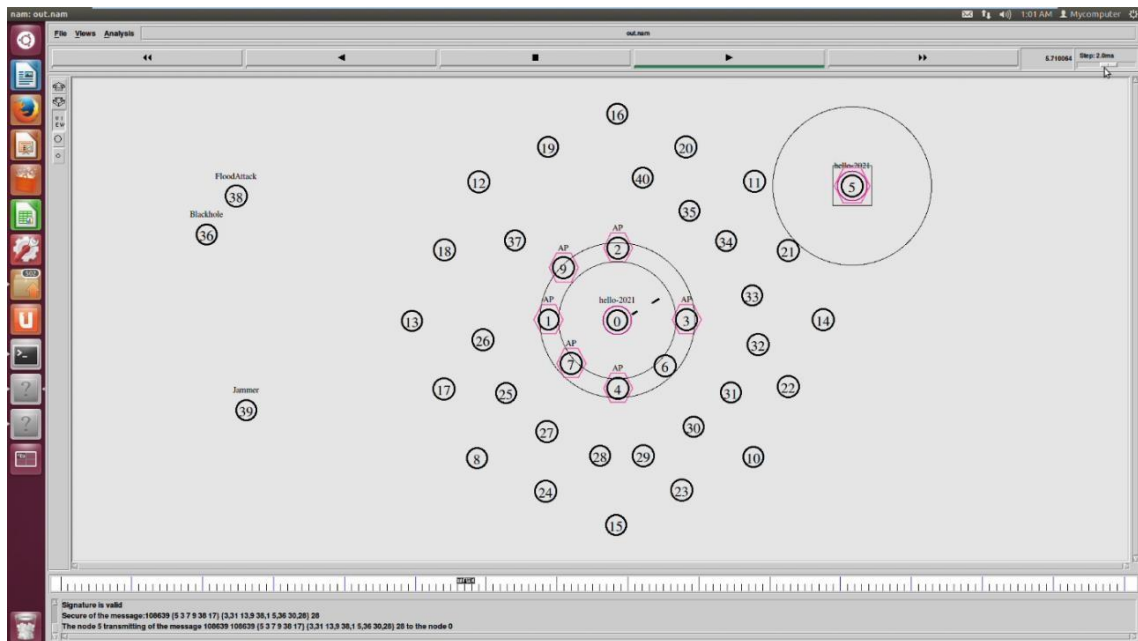


Fig. 5.5 Receiving data at Receiver node from transmitter node

5.8 ECC – NAM RESULTS

Algorithm : ECC (Elliptical Curve Cryptography)
 Input data : hello-2021
 Transmitter Node : 5
 Receiver Node : 1
 Total Nodes : 41

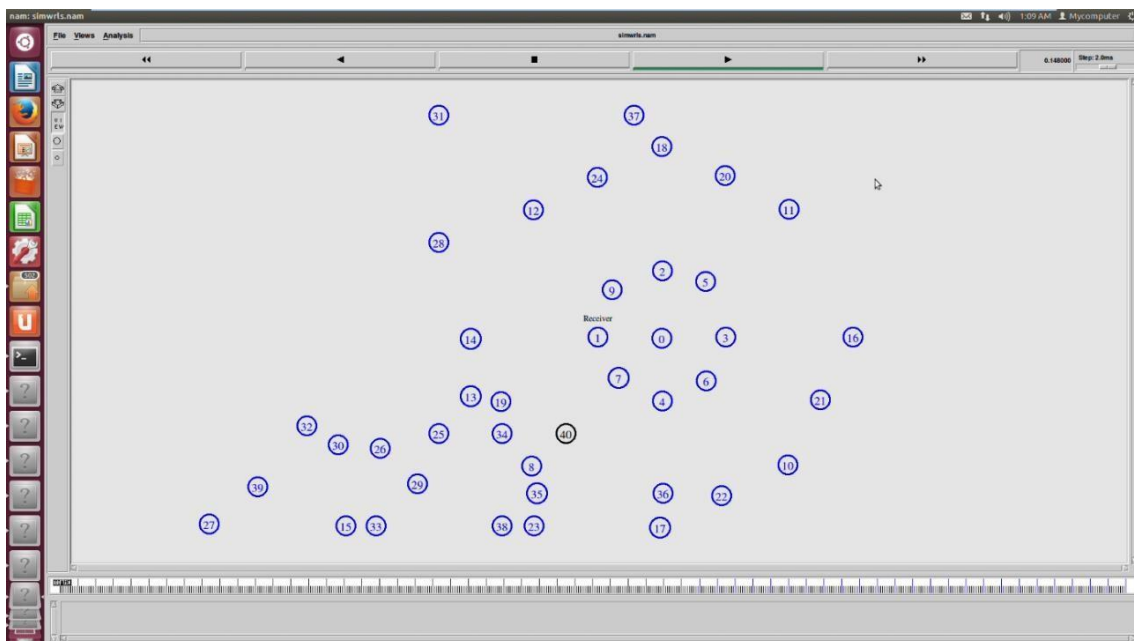


Fig. 5.6 Network Representation NAM output (ECC)

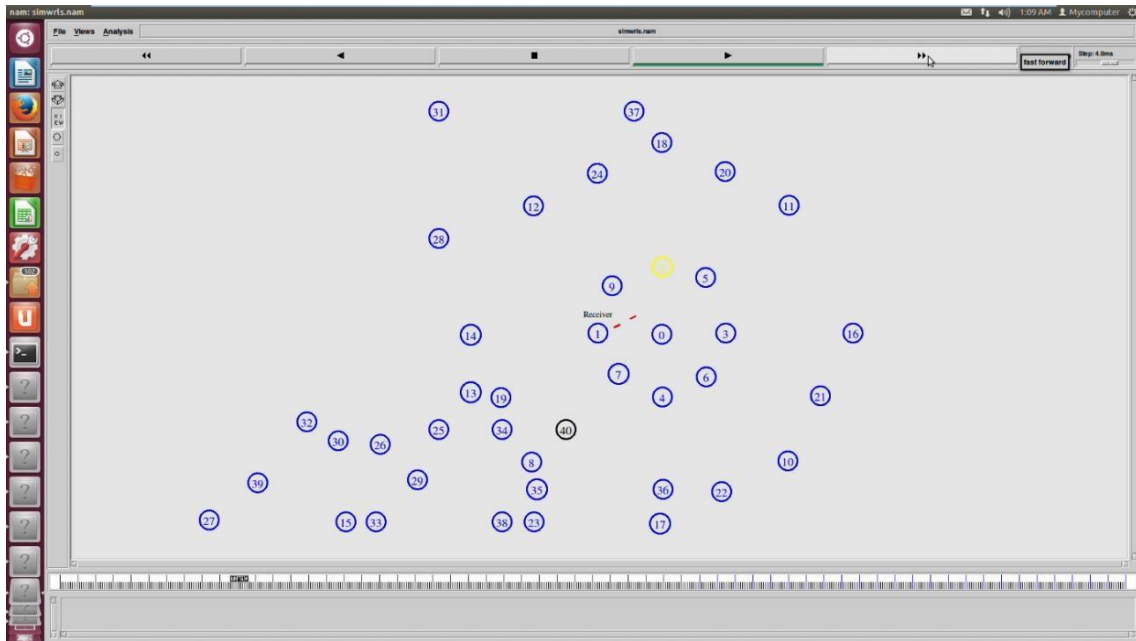


Fig. 5.7 Data transmission between Transmitter and Receiver nodes

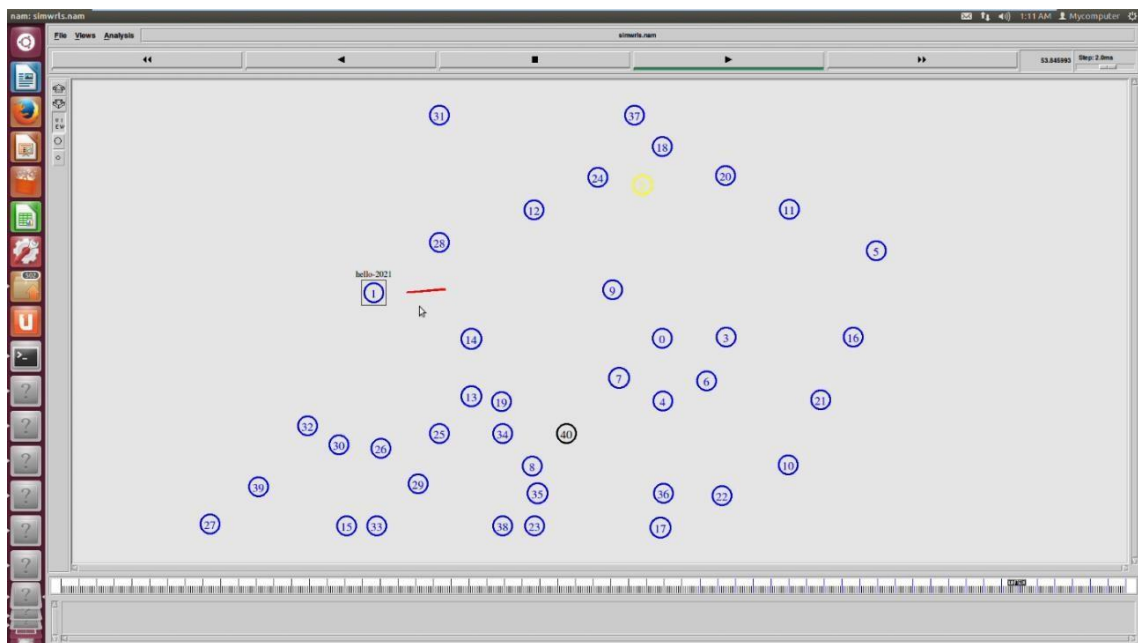


Fig. 5.8 Routing path between Source and Destination

5.8 **PARAMETER ANALYSIS**

5.8.1 ***RSA Results***

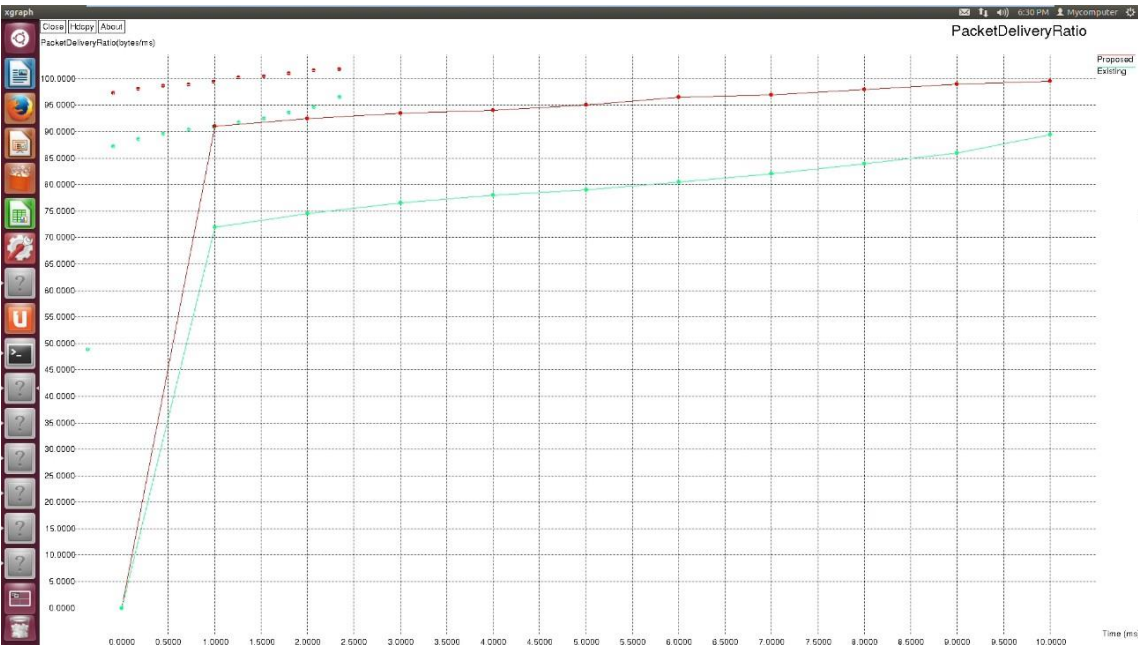


Fig.5.9 Packet Delivery Ratio

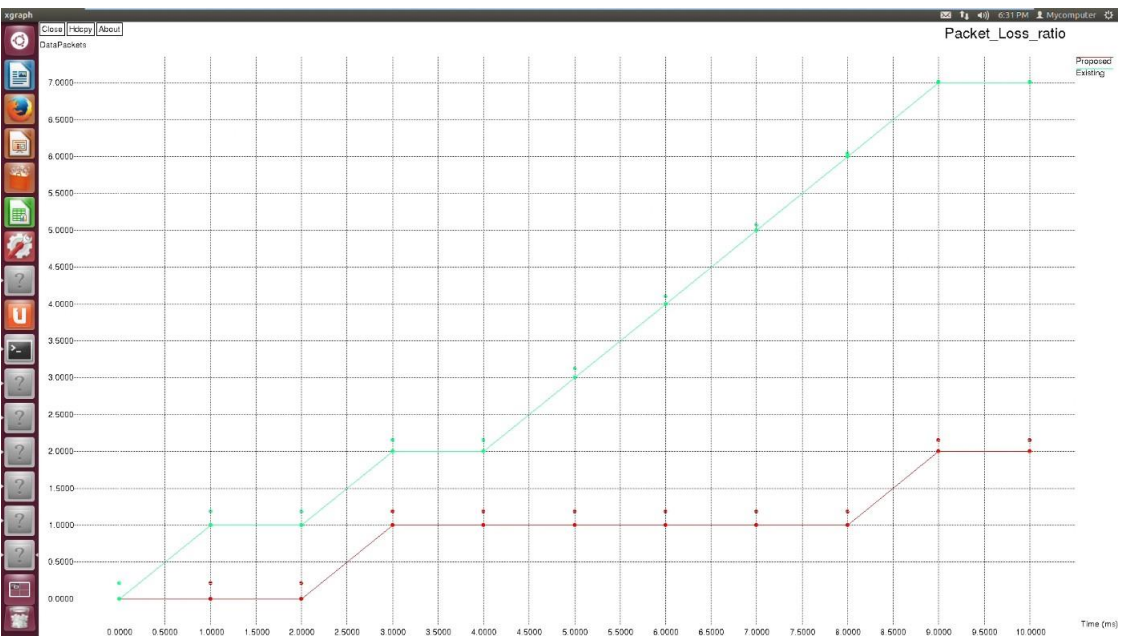


Fig. 5.10 Packet Loss Ratio

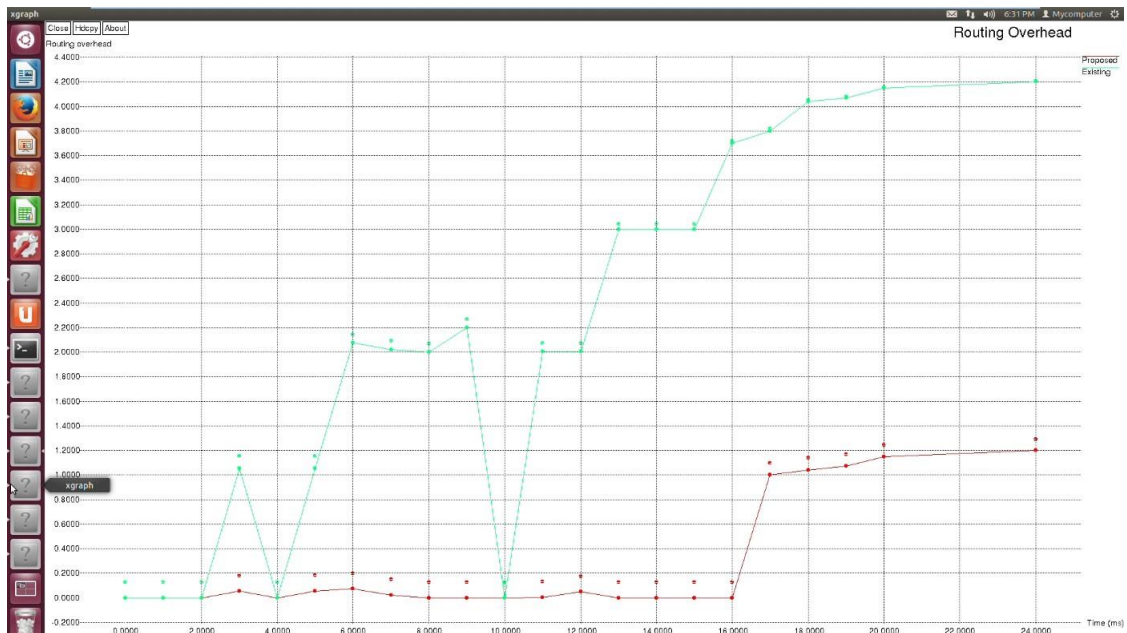


Fig. 5.11 Routing Overhead



Fig. 5.12 Comparison Of Energy

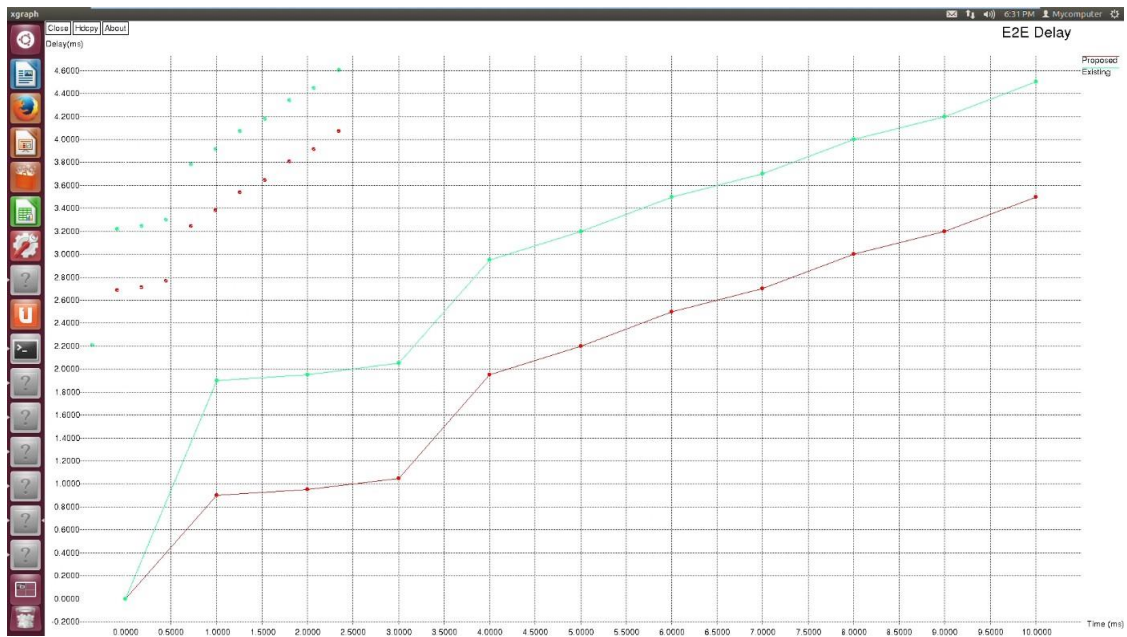


Fig. 5.13 End to End Delay

5.9.2 ECC Results

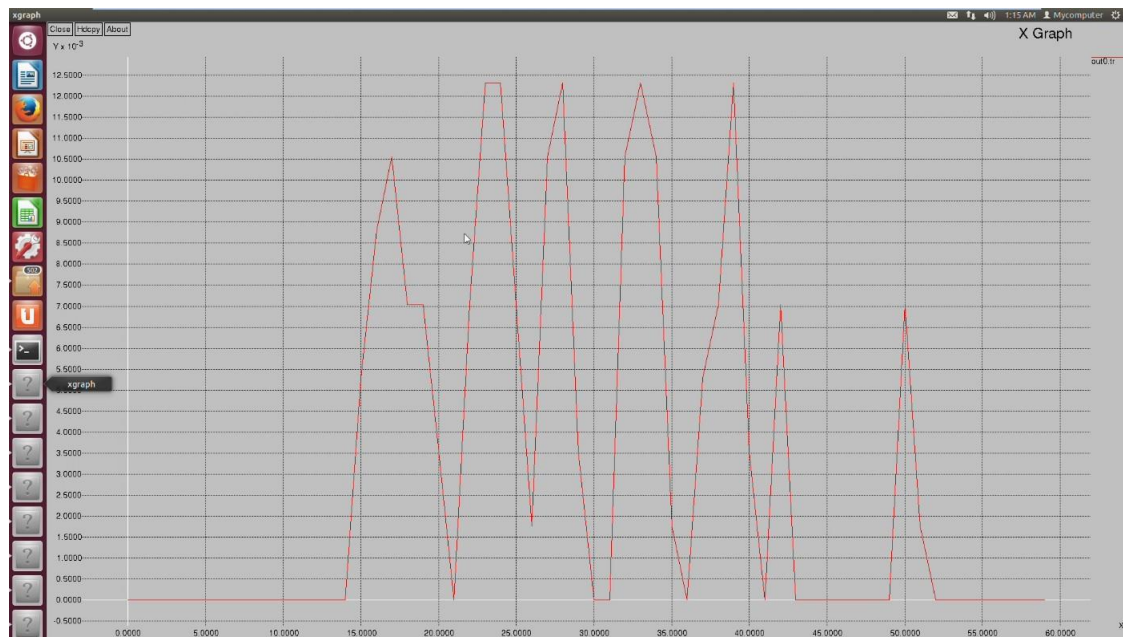


Fig. 5.14 Network Performance of ECC

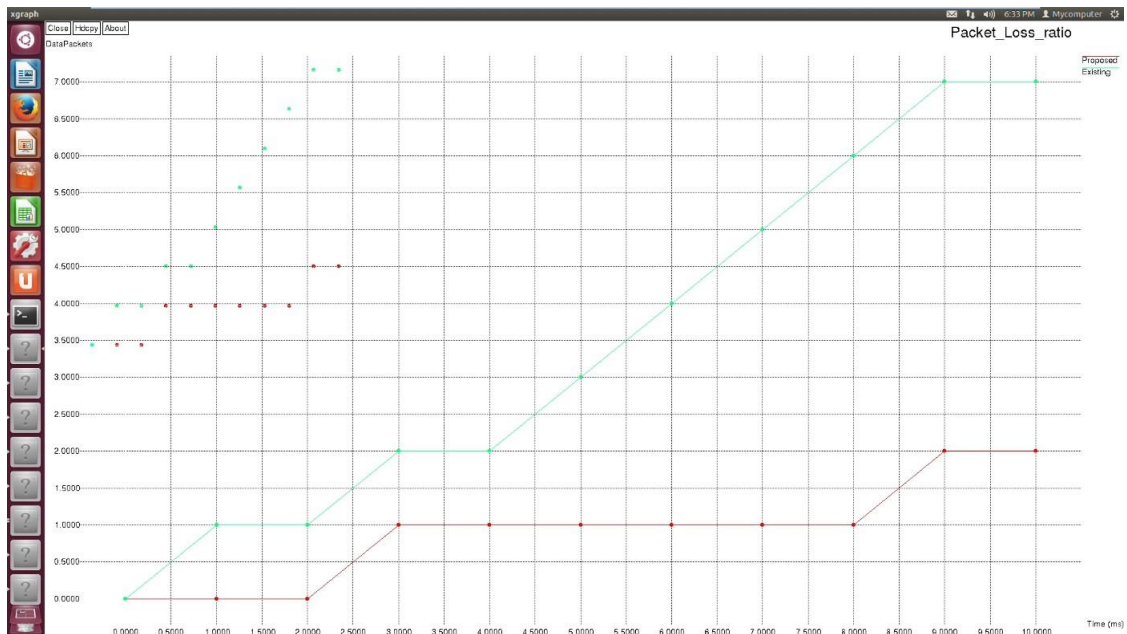


Fig. 5.15 Packet Loss Ratio

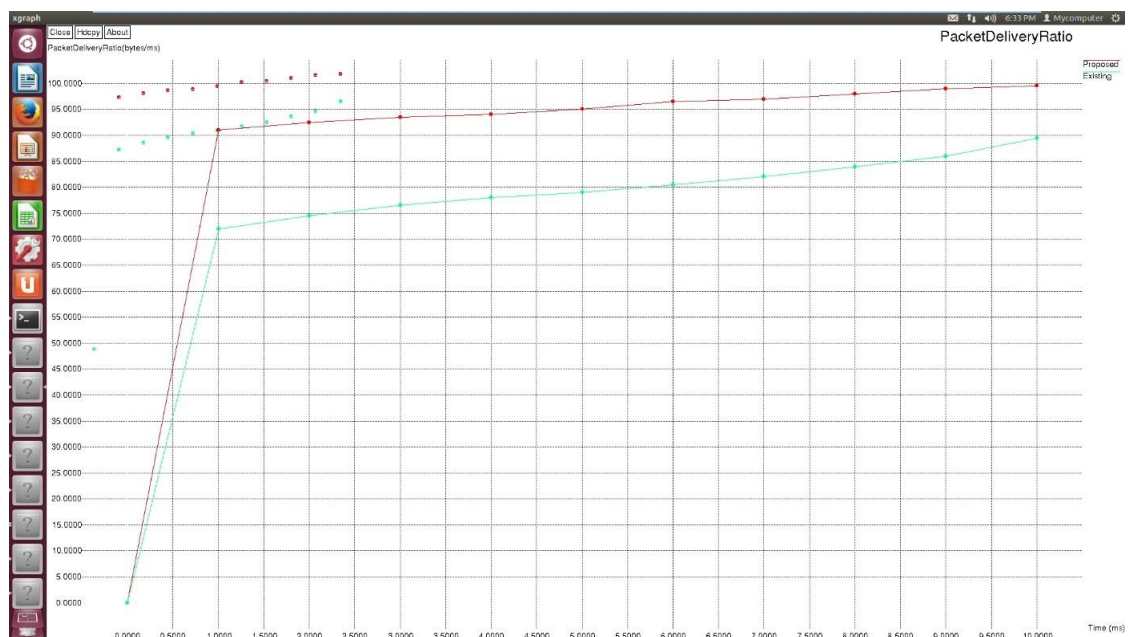


Fig. 5.16 Packet Delivery Ratio

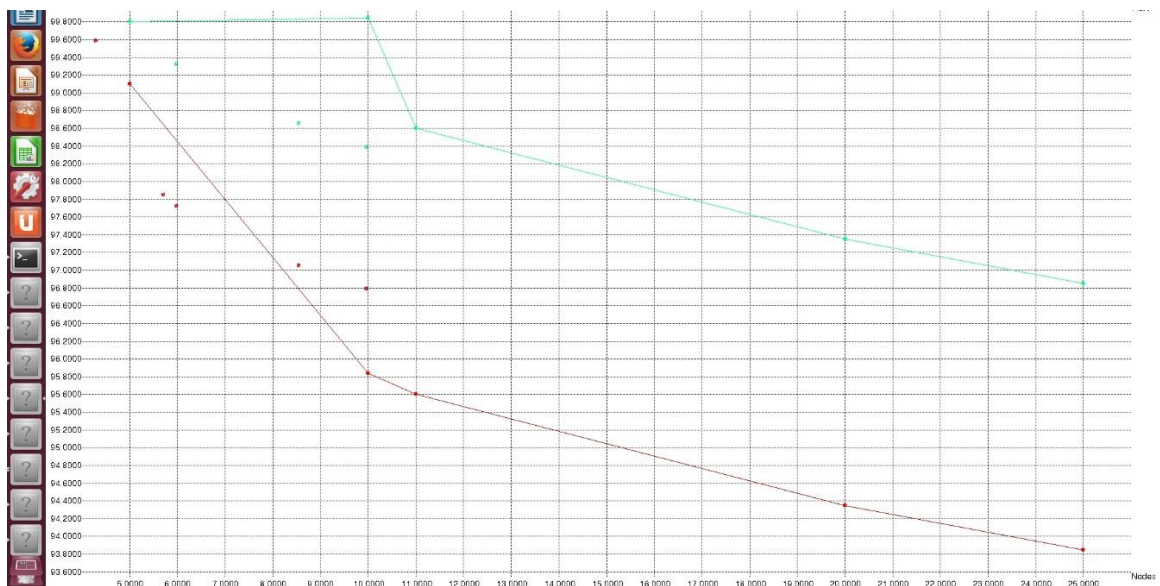


Fig. 5.17 Comparison of Energy



Fig. 5.18 Routing Overhead

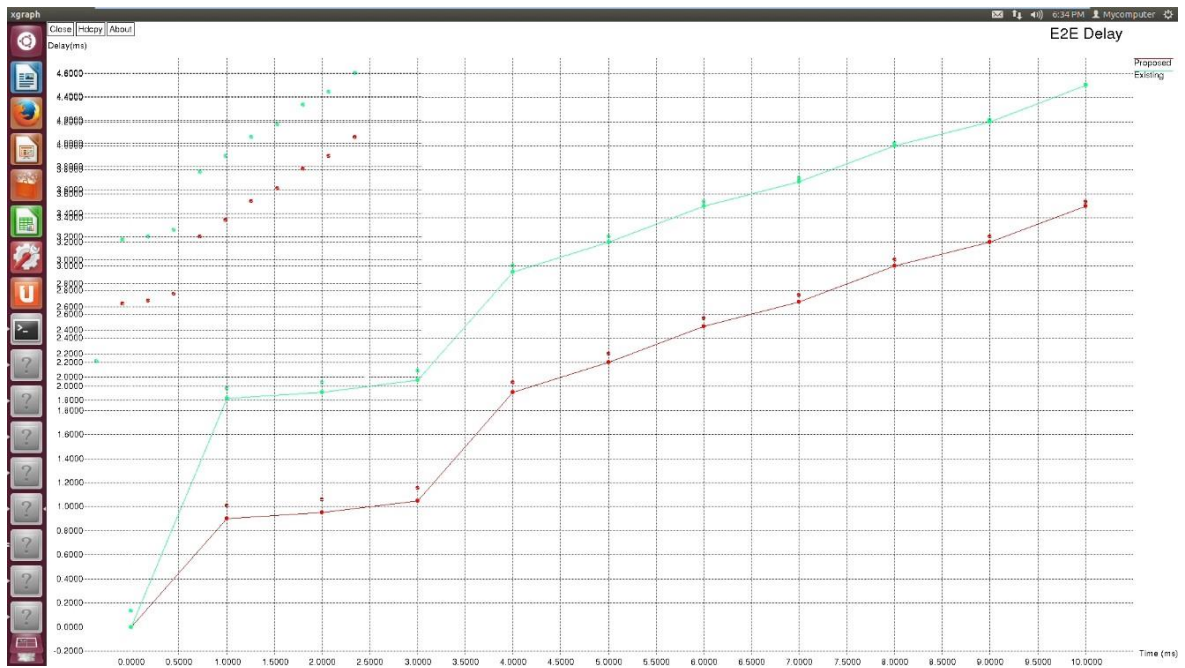


Fig. 5.19 End to End Delay

CHAPTER 6

SUMMARY AND CONCLUSION

Getting basic information move about the climate in military applications are essential since such information aid dynamic interaction. Late secure methodologies should be followed for moving the information. The got approaches utilized for moving the top-k question bring about the proposed work help with moving the basic information's safer in the multitier design of WSN. It in this way brings about accomplishing the outcome uprightness and fulfillment confirmation which is fundamental in numerous sensor related applications specifically modern non military personnel zones, territory checking and climate observing.

REFERENCES

- [1] Al-Fuqaha, A., Guizani, M., Mohaßmmadi, M., Aledhari, M. and Ayyash, M. "Internet of things: A survey on enabling technologies, protocols and applications," in IEEE Communications Surveys Tutorials, vol. 17, no. 4, 2015, pp. 2347-2376.
- [2] Brandt A. and Buron, J. "Transmission of IPv6 Packets over ITU-T G.9959 Networks," IETF RFC 7428, February 2015, <https://www.ietf.org/rfc/rfc7428.txt>, (accessed February 24, 2017).
- [3] Cetinkaya O. and Akan, O. "A dash7-based power metering system," in 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015, pp. 406–411. [15] HomePlug Alliance, "HomePlug™ AV2 Technology," 2007 http://www.homeplug.org/media/filer_public/2c/32/2c327fc8-25bb409e-abf7-c398534c24dc/homeplug_av2_whitepaper_130909.pdf, (accessed February 24, 2017).
- [4] Decuir, J. "Bluetooth 4.0: Low Energy," 2010, <https://californiaconsultants.org/wp-content/uploads/2014/05/CNSV-1205-Decuir.pdf>, (accessed February 24, 2017).
- [5] Gomez, C., Oller, J. and Paradells, J. "Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology," in Sensors, vol. 12, no. 9, 2012, pp. 11734-11753.
- [6] Granjal, J., Monteiro, E. and Sa Silva, J. "Security for the internet of things: A survey of existing protocols and open research issues," in IEEE Communications Surveys Tutorials, vol. 17, no. 3, 2015, pp. 1294-1312.
- [7] Hasan, M., Hossain, E. and Niyato, D. "Random access for machine- to-machine communication in lte-advanced networks: issues and approaches," in IEEE Communications Magazine, vol. 51, no. 6, 2013, pp. 86-93.
- [8] IEEE802.15.4-2011, "IEEE standard for local and metropolitan area network—part 15.4: Low-rate wireless personal area networks (LR-WPAN)," in IEEE Standards, April, 2012, pp.1-225.
- [9] ITU-T, "Short range narrow-band digital radio communication transceivers PHY and MAC layer specifications," 2012, <https://www.itu.int/rec/T-REC-G.9959>, (accessed February 24, 2017).

- [10] Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F. and Alonso- Zarate, J. "A survey on application layer protocols for the internet of things," in Transaction on IoT and Cloud Computing, vol. 3, no. 1, 2015, pp. 11-17.
- [11] Kim, A., Hekland, F., Petersen, S. and Doyle, P. "When hart goes wireless: Understanding and implementing the wirelesshart standard," in IEEE International Conference on Emerging Technologies and Factory Automation, 2008, pp. 899–907.
- [12] Nobre, M., Silva, I. and Guedes, L. A. "Routing and scheduling algorithms for WirelessHART Networks: a survey," in Sensors 15, no. 5, 2015, pp. 9703-9740.
- [13] Park, M. "IEEE 802.11ah: sub-1-ghz license-exempt operation for the internet of things," IEEE Communications Magazine, vol. 53, no. 9, 2015, pp. 145-151.
- [14] Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., McCann, J. and Leung, K. "A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities," in IEEE Wireless Communications, vol. 20, no. 6, 2013, pp. 91-98.
- [15] Stamford, "Gartner's 2014 hype cycle for emerging technologies maps the journey to digital business," August 2014, <http://www.gartner.com/newsroom/id/2819918>, (accessed February 24, 2017).
- [16] Zigbee, "Zigbee resource guide," 2016, http://www.nxtbook.com/nxtbooks/webcom/zigbee_rg2016/#/0, (accessed February 24, 2017).
- [17] Z-Wave, "Z-wave protocol overview," April 2006, https://wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243_2_Z_Wave_Protocol_Overview.pdf, (accessed February 24, 2017).

RANDOMIZED GENERATION TECHNIQUE FOR SECURED DATA TRANSMISSION BY CRYPTOGRAPHIC MECHANISM

Submitted in partial fulfilment of the requirements for the award of
**Bachelor of Engineering Degree in
Electronics and Communication Engineering**

By

PATHAKAMURI SRINIVAS (37130302)
NUVVULA NAGA TARUN KUMAR (37130290)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PATHAKAMURI SRINIVAS (REG.NO 37130302)** and **NUVVULA NAGA TARUN KUMAR (REG.NO 37130290)** who carried out the project entitled "**RANDOMIZED GENERATION TECHNIQUE FOR SECURE DATA TRANSMISSION BY CRYPTOGRAPHIC MECHANISM**" under my supervision from AUG 2020 to MARCH 2021.

Internal Guide

Ms. YOGALAKSHMI

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **PATHAKAMURI SRINIVAS (37130302)** and **NUVVULA NAGA TARUN KUMAR (37130290)** hereby declare that the Project Report entitled “**RANDOMIZED GENERATION TECHNIQUE FOR SECURED DATA TRANSMISSION BY CRYPTOGRAPHIC MECHANISM**” done by us under the guidance of **Ms. YOGALAKSHMI**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1.

DATE : 17.04.2021

PLACE : CHENNAI

2.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N.M. NANDITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Ms. YOGALAKSHMI**. Asst. Professor, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

This article explores many issues relating to the Wireless Sensor Network (WSN) and the relevance of Elliptic Curve Cryptography (ECC). WSN security is more noteworthy in WSN, given that sensor hubs and remote connections are impeded by preparation. Widespread use of WSNs leads to different kinds of hazards. Adequate protection plans are required to protect against the dangers. Customary authentication is done by machinery or programming and generally carried out by means of encryption techniques. Limited area, existence of connections, restricted handling, force and WSN memory imply exact restrictions on cryptographic method determination. This paper looks at the data transfer over IOT and the two modified RSA and elliptical curve cryptography (ECC) cryptography calculations are the best applicants for their smaller scale. Despite the smaller key size, high security creates regional and efficient crypt structures.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	xi
1.	INTRODUCTION	1
1.1	METHOD OF APPROACHING	3
2.	LITERATURE SURVEY	5
2.1	COMPARISON OF LITERATURE SURVEY	9
3.	AIM AND SCOPE	10
3.1	EXISTING SYSTEM	11
3.2	PROPOSED SYSTEM	11
3.3	NEIGHBOR DISCOVERY BLOCK	12
3.4	BROADCASTING NODES	13
4.	ALGORITHM AND IMPLEMENTATION	14
4.1	ALGORITHMS	14
4.1.1	ECC ALGORITHM (ELLIPTICAL CURVE CRYPTOGRAPHY)	14
4.1.2	RSA ALGORITHM (RIVEST SHAMIR ADLEMAN)	17
4.2	IMPLEMENTATION	19
4.2.1	OTCL ESSENTIALS	20
4.2.2	ASSIGNING ESTEEMS TO FACTORS	20
4.2.3	PROCEDURES	21
4.2.4	FILES AND RECORDS	22

CHAPTER NO.	TITLE	PAGE NO.
	4.2.5 CALLING SUBPROCESSES	22
	4.2.6 CREATING THE GEOGRAPHY	23
	4.2.7 NODES	23
	4.2.8 AGENTS, APPLICATIONS AND TRAFFIC SOURCES	24
	4.2.9 TRAFFIC SINKS	25
	4.2.10 LINKS	26
	4.2.11 TRACING AND CHECKING	28
	4.2.12 TRACES	28
	4.2.13 MONITORS	30
	4.2.14 CONTROLLING THE RECREATION	30
4.3	SIMPLE STIMULATION EXAMPLE	31
4.4	EXTENDING NS: CREATING ANOTHER AGENT	37
4.5	THE HEADER RECORD	37
	4.5.1 TABLE OF MYPING.H	40
	4.5.2 TABLE OF MYPING.CC	44
4.6	NECESSARY CHANGES IN NS	44
5.	RESULTS AND DISCUSSION	45
5.1	EDIT PACKET.H FILE	45
5.2	EDIT NS-PACKET.TCL DOCUMENT	46
5.3	EDIT NS-DEFAULT.TCL	47
5.4	EDIT MAKEFILE.IN	47
5.5	RECOMPILE NS	48
5.6	SIMPLE EXAMPLE USING MYPING AGENT	48
5.7	RSA - NAM RESULTS	49
5.8	ECC - NAM RESULTS	51
5.9	PARAMETER ANALYSIS	53

CHAPTER NO.	TITLE	PAGE NO.
	5.9.1 RSA RESULTS	53
	5.9.2 ECC RESULTS	55
6.	SUMMARY AND CONCLUSION	59
	REFERENCES	60

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
4.1	FORWARD ROUTE DISCOVERY (ECC)	16
4.2	BACKWARD ROUTE DISCOVERY (ECC)	17
4.3	KEY GENERATION IN ECC	17
4.4	LINK IN NS2	27
4.5	LINK IN NS2 WHEN TRACING IS ENABLED	29
4.6	A SIMPLE NETWORK TOPOLOGY AND STIMULATION SCENARIO	32
4.7	NS DIRECTORY STRUCTURE	44
5.1	NETWORK REPRESENTATION NAM OUTPUT FILE (RSA)	49
5.2	AUTHENTICATION BETWEEN NODES	49
5.3	VALIDING SIGNATURE (KEYS) IN NETWORK	50
5.4	ROUTING IN NETWORK OVER OFF THE ATTACKERS	50
5.5	RECEIVING DATA AT RECEIVER NODE FROM TRANSMITTER NODE	51
5.6	NETWORK REPRESENTATION NAM OUTPUT (ECC)	51
5.7	DATA TRANSMISSION BETWEEN TRANSMITTER AND RECEIVER NODES	52
5.8	ROUTING PATH BETWEEN SOURCE AND DESTINATION	52
5.9	PACKET DELIVERY RATIO	53
5.10	PACKET LOSS RATIO	53
5.11	ROUTING OVERHEAD	54
5.12	COMPARISON OF ENERGY	54
5.13	END TO END DELAY	55
5.14	NETWORK PERFORMANCE OF ECC	55
5.15	PACKET LOSS RATIO	56

FIGURE NO.	TITLE	PAGE NO.
5.16	PACKET DELIVERY RATIO	56
5.17	COMPARISON OF ENERGY	57
5.18	ROUTING OVERHEAD	57
5.19	END TO END DELAY	58

LIST OF ABBREVIATIONS

BPA	-	Basic Probability Assigning
ECC	-	Elliptical Curve Cryptography
FIFO	-	First-In First-Out
MR	-	Magnetic Resonance
QoS	-	Service Quality
RSA	-	Rivest Shamir Adleman
WSAN	-	Wireless Sensors and Actuator
WSN	-	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

- Today, with the rapid advancement of communication domains, instant delivery and arrival of data over a network or different networks is possible. But along with it much emphasis is given on the security of data during its traversal over the network.
- Data can be identified, modified and even destroyed by unauthorized users during its traversal. So, the solution to this is to transmit encrypted data over the network so that it is impossible to identify the message by the hands of the unauthorized users.
- Cryptography is an art and science of mathematically scrambling the data so that only someone with the necessary key can unscramble it.
- It contains mainly two domains i.e. Encryption and Decryption.
- Web of stuff (IoT) is a first invention that focuses on the connecting of stuff or gadgets and people or customers to achieve such common aims. IoT is powered by many current developments, such as wireless sensors and actuator (WSAN) and identifiable evidence of radio recurrence (RFID). Kevin Ashton from Auto ID-Center MIT first introduced the possibility of Internet of things. Due to the widespread accessibility of the Web as Wi-Fi (3G, 4G LTE), the comprehensive detection has become evident from now on. Thus, an entrance was opened to people and customers who will finally be adding to the shrewd urban areas in future. The amount of gadgets associated with this would increase enormously and reach somewhere between 50 and 100 billion between now and 2020. This enormous number of gadgets can detect and make administrations widely available. The data and correspondence systems

are constantly integrated in the IoT worldview into our environment. So detect and prepare different true questions and place the data on distant mist. IoT is a key element in the development of lively homes, skilful cities and a brilliant infrastructure for medical services. IoT is widely recognised on the opportunity to gain customer confidence by providing strong security and defence. IoT security is one of today's hot explorations. Many analysts worldwide use their efforts to resolve various challenges in IoT security. However, because of its heterogeneity, protection IoT is an exceptional examination. These improvements have their very own standard security and safety defects, which appear to be found in IoT settings, as the web of things is a combination of these innumerable inventions. In this document, the IoT architecture for security evaluation on each IoT layer is briefly discussed. We speak about safety hazards in each layer and possible attacks by enemies. We also suggested counter-measures to alleviate the risk of these attacks. IoT's base is exceptionally inclined towards significant security attacks, such as Denial of Service (DoS), replay assaults, centre station guy, cloning stuff, snoops and steering attacks. For eg, gadget hardening, security failure, data revelation, DoS and spoofing, signal infusion and side channel attack have been explicitly assembled by Atamli et al. Atamli. IoT devices are compulsory and existing cryptographic safety arrangements cannot be used for these gadgets, making them inclined to reliability and confidentiality of information.

- Similarly, three Security Goals such as confidentiality, rectitude and accessibility are difficult to achieve with composition work assaults. This paper analyses subtleties on the problems of traditional safety structures in IoT. Additionally a major IoT protection problem is the validation and access control mechanism of gadgets. The huge number of gadgets and machine to machine (M2M) correspondence existence of IoT are the reason for the verification and access control problems in IoT. A double reading of medical pictures can improve the detection of the tumour.

1.1 METHOD OF APPROACHING

Many new conventions are normalised every year with the irregular flow of examination interest for IoT. Overview papers are therefore continually written to feature different IoT-identified aspects of normalisation. Instances in such papers recall the IETF instructions, safety conventions and layer principles as well as applications or transports. Different articles discuss a specific layer of standards, such as correspondence agreements or guidance. Summarize in particular the core norms provided by different organisations of guidelines until 2015. It also discusses various IoT problems such as flexibility and adaptability. In this paper, we hope to provide a detailed analysis of new concepts, draughts and conventions which broaden the work that has been done. This allows us to explore further principles, introduce a section of the new standards in the IETF and discuss the cutting-edge agreements on which normalisation will soon take place. The As illustrated in, the IoT environment of the ecosystem includes a seven-layer model: sector, secured, interconnected, incorporated, examined, applied and administrative. The layer on the base could be a genius system, a home-related framework, or shrewd wellness, and so on. Sensors and skilled gadgets that can be called the application centre will be included in the next layer. The form and distribution of the sensors varies according to the ideal applications. Temperature sensors, mugginess sensors, power metres or cameras are instances of such sensors. The third layer consists of an interconnection layer which promotes sensor information correspondence to a server farm or cloud.

In addition to other information sets such as geology, population or monetary information, the information is included. The combined knowledge is also analysed using IA and IT strategies. Coordinated effort and correspondence programming in new application levels can facilitate these enormous distributed applications. These principles include organisational programming (SDN), engineering administration (SOA) and so on. Last but not least, the top layer includes administration results such as energy management, board well-being, preparation, transportation etc. For each one of these 7 layers based on each other, security and the board are essential and thus they appear as a background.

The recent study showed how supervised techniques such as the Convolution Neural Network allow for the classification of human brain in magnetic resonance (MR) images. The approach proposed would dramatically reduce the time needed to diagnose and classify the brain tumour. Increasing the detection efficiency and the entire system dramatically reduces tumour detection time and expense. The main objective of the device proposed is that it can be used by anyone due to its decreased cost for the detection process.

CHAPTER 2

LITERATURE SURVEY

Kamil Samara has suggested energy efficiency, load control and a wireless network routing protocol. The led diffusion algorithm saves power by hop-transference of data packets and flood prevention routes. The goal is not to find the best or the most healthy routes in the directed diffusion algorithm (healthier paths are paths that utilise less total energy than others and avoid critical nodes). The quality of energy use can be increased by the use of healthier roads. He suggested an ideal WSN routing protocol based on the total energy available on the route, running time and avoiding critical nodes to ensure that the safe routes are respected. This decision is taken with information about possible routes, and with non-incremental computer planning, we introduce paths that connect our parameters. Small networks did not show a substantial difference between the selected protocol and the specified algorithm for the simulation performance. The performance of the project protocol has increased as the network size increases.

Cumar Pradeep ATREY² implemented reliable, safe data collection in Puneth and Muralidhar Kulakarni ¹'s wireless sensor networks. By randomization and dispersal routing, the feature compensates for H-SPREAD deficiencies. This method enhanced security with unequal dispersion compared to H-SPREAD. The network lifecycle and security are all taken into consideration for an unpredictable and deterministic combination. As a simple implementation process, i.e. nodes (circular) and BS, the approach was very specific (center of the network area). The research looks at the SSR architecture, which allows for secure, energy efficient wireless networks for group communications. This method increases energy efficiency, but does not manage the assault by CN. They have high security and cn attack is not feasible. They have high protection. This process is taking place.

As an alternative to the Wireless sensor networks with handheld dishes, Jesse Whitehead suggested cluster-based confidence proliferation and energising data collection. When it is not possible to find vital and engaged core stations such as tornado or volcanic eruption, mobile sinks should be given to capture data. In order to link nodes securely on the basis of confidence and network clusters, Jesse Whitehead suggested a smooth clustered structure. Jesse Whitehead, the emulation display, shows that the energy conservation and networking life of this project exceeds anyone.

Wireless network sensor management had been implemented by Ahmed Khana, Hassaan Khaliq Qureshia and Adnan Iqbalb Energy. Wired batteries are also used to power the nodes. Any battery powered network service energy saving devices are implemented. The limitation linked to the low power of the battery forced the research trend to find new sources through environmental collection. This study determines a high degree of classification in WSNs for energy efficiency. They list a number of schemes focused on battery power and energy collection schemes. The most recent change to the wireless power transmission in the sensor node is also illuminated as an alternative to conventional batteries. Alternatively, they proposed the conventional energy savings that the sensor network would make when designing energy efficient systems, taking account of recent energy allotments.

In order to collect positive and enthusiastic data on unused wireless networks, Farah Kandah, Jesse Whitehead and Peyton Ball launched it. It is important that important degradation of the system caused by incorrect or inaccurate transfer of data be ignored. Building trust - assurance between two devices guarantees the confidence and accuracy of data, providing a safe mechanism to identify all devices confidently and authorising no particulars in uncertified user details. The output showed that the impact of malicious nodes on the network must be isolated or reduced as the number of loops decreases and therefore decreases energy consumption in the network and increases its lifespan. The benefit of clustering is also the distribution of energy demand across nodes, which prevents node drop and network difficulties at an early stage.

Guosheng Huang has generated a number of energy-efficient data for unmanned aircraft. The maximised trajectory algorithm ant-colony (UAV), which forms the fewest possible data anchors in the field of operation, is given. Secondly, trust and process assessment schemes will be used to identify the trust rating of sensor nodes and only credible data to certify the quality of data collection will be collected. Finally we will find an effective data collection process that would contribute to achieving a much better network power consumption in our TEEDC-UAV scheme, which has been tested on a wide range of experimental output. The network life has increased by 48.9 percent relative to previous approaches. Meanwhile, the trust feature in this article will improve the established precision of a degree of node trust that affects 91 percent of the life of the network.

For the wireless sensor networks, Khalid Haseeb created a heuristic protocol. The proposed protocol allows for heuristic surveys on artificial intelligence to resolve the need for a system for dependence and smart learning. Secondly, transmissions with the least complexity compared with opponent classes are protected for protection. Traffic exploration can help to reduce the loss and network disconnectivity of the pathways management approach. Simulation results indicate that the SEHR protocol increases the efficiency of the network performance by average of 18%, a 42% packet reduction rate, at 26%, a 36% energy consumption of faulty routes, and a 43% numerical rate compared to current work.

Naveed Islam implemented the Lightweight Safe and Efficient Sensor Network Routing Protocol. This work enhances the energy efficiency, utilises the Service Quality (QoS) factor and makes time sensitive applications at network edges more accessible. This protocol is centralised on two primitive layers of encryption, ensuring data in real-time. Cluster heads and fog nodes have a lightweight data confidentiality system first, and an asymmetric high performance framework is introduced to define the layers of cloud and fog. In terms of routing, reliability and network control, the survey of simulation-based research has found important results in relation to current technologies.

Liangrui Tang, Zhilin Lu and Bing Fan suggested that the wireless sensor network routing algorithm be energy efficient and reliable. DS-EERA creates three

guides as residual electricity, traffic, and so forth. They gain entropy weight to monitor the weight of three guides objectively. The DS proof theory fusion rule registers to fuse each index value with the BPA function to pick a next hop after the intent of the basic probability assigning (BPA). Each network node transfers data via this routing approach. DS-EERA is a very promising theoretical paper and simulation result, enabling it to extend its network life effectively. In the meantime, the loss rate of packets can also be reduced and data transfer reliability increased.

Muhammad Ilyas suggested an energy-efficient protocol for Internet routing for trust-based sensor networks. They also implemented a three-tier clustering system with an integrated authentication function to recognise and blacklist malicious sensor nodes. Clustering protocol with the sinknote used for the cluster head and grid head on the same priority. For testing link performance and enhancing routing reliability, hardware-based link level estimators are used. In the end of the day, the effectiveness of the introduced technique was assessed by unnecessary experiments. It exceeds many related protocols such as the futile logic of the hybrid ants-based clusters and colony-based optimization.

A survey on data collection and targeted methods of energy-efficient routing in WSN has been introduced by U.Nilabar Nisha, S.Velmurugan, L.Raja, G.Shanthi. However, energy usage did not decrease and network lifetime was not improved in working methods during the processing, data collection and goal surveillance. Our main goal is to reduce energy consumption and the reliability of data collection in WSN data collection and monitoring. In this study, a survey is carried out using methods to achieve greater energy efficiency and maximum efficiency in data collection with a minimum target monitoring time in WSN.

Venkateswara Rao M has implemented a Zone based Wireless Sensor Network Safe Energy Efficient Attack Resilient Process. This work provides a secure, energy-efficient, and aggressive routing solution for zone-based wireless sensor networks including proactive monitoring and malicious attack prevention. This study divides the network into areas and gives the profit rate of the spreading packet across the region, a probabilistic and fused score that differs from previous studies in the observation of any malignant node. The routing can be

changed to accommodate continuing treatments for energy and protection. A strong evaluation that can influence the network operations cannot determine the interaction of the systems over the life cycle of the network. In NS2, the solution applied is checked and work alternatives compared to demonstrate the effectiveness of the method.

2.1 COMPARISON OF LITERATURE SURVEY

AUTHOR	PROBLEM ADDRESSED	METHODOLOGY PROPOSED	REMARKS
Kamil Samara (2020)	Routing Algorithm	Direct diffusion algorithm	Simulation result did not point out a significant difference in smaller networks
Deepa Puneeth, Nishanth Joshi, Pradeedp Kumar Atrey, Muralidhar Kulakarni (2020)	Security	H-Spread	In this method we can achieve low security
Jesse Whitehead (2020)	Time Management	Cluster based mechanism	Inference high performance
Junaid Ahmed Khana, Hassaan khaliq qureshine, Adan Iqbalb (2019)	Energy Management	Battery driven energy consumption	High performance
Farah Kandah ¹ , Jesse Whitehead ² , Peyton Ball.(2019)	Energy consumption	Reducing number of nodes	High performance

CHAPTER 3

AIM AND SCOPE

- This paper mainly uses the remote sensor network based on the occasion divulgation and data range. The hubs often communicate information to the base station via multi-jumps due to their restricted correspondence gap between the sensor hubs.
- Instructions for finding a way to guide calculations with lowest energy use from a source hub to a sink hub. Figure 1 shows the geography of WSN. By multi-bounce sending of the collected data, the data assortment terminals are viewed as the fundamental hub.
- Assume there are several regular hubs and a lonely sink hub in the centre in a sensor network with a territory $L = L$, and not all hubs are, shifted by sending.
- The zone between the hubs is fixed. Both sensor hub systems are isomorphic, have capacities to steer, send and accept, two hub can be transmitted in single or multi-bounce mode and the subordinate state is very similar.
- The heap of hubs underneath is 0, and energy underlying is E_0 . The hubs trading data approach is the same. Each hub has an interesting ID, supporting the storage of data such as remaining power, package ID, next bounce ID, sender ID and so on. These data are constantly refreshed as the next person changes.
- It is noted that hubs have been limited in traffic lengths and that a first-in first-out (FIFO) package of information is prepared.

3.1 EXISTING SYSTEM

In view of its easy scaling and multisectoral characteristics, the promotion of competent remote communications and the advance of electronic data innovation (WSN) is widely used in various fields. Nonetheless, the WSN hubs are typically battery powered and generally shipped outside or under harmful conditions, making them poorly equipped for energy regeneration. The cost of excess shipment and replacement of hub is typically high. A efficient steering procedure is therefore supposed to restrict the use of the network's resources and take lifetime. Since the power of the sensor hub is mostly used for the collection and transfer of information, the traditional directions approach mainly thinks how to use the restricted means of moving information from the source hub to the target as quickly as possible. In any case, a lot of information is transmitted via a 'Many-to-one mode' from the source hub in the energy-induced sensor organisation, with genuine 'Channel Effects' and 'Energy Opening' concerns being effective.

3.2 PROPOSED SYSTEM

The "many-to-one" mode of communication can also cause the organisation to block its data. For instance, when a main occasion is set, the hubs must send a lot of information in a short time to the sink hub. Clog will dispose of a huge number of information packs, reducing the unequivocal efficiency of data transfer. In the meantime the organisation, because of data retransmissions that reduce energy consumption, would produce superfluous energy utilisation and an increased transmission time. Some blockage scans, when blocking is discovered, usually use such acts to monitor the inflow speed of information, and it is difficult for the upstream hub to manage the rate of information flowing upstream from blocking. Many steering systems often like hubs to reduce clogs, but because of diversions the hubs can lose more energy. Therefore we propose an energy efficient and solid guiding calculation based upon ECC, by receiving multi-characteristic dynamic technique, to enhance efficiency of energy and to increase organisational lifetime and reliability of information transmission. This paper has the main commitments as follows:

Proposed Routing Model constructs the centre evaluation work from the perspective of the reduction in energy use. This means that the hubs' a scripts are fully considered and disconnected into three lists: the energy transmission proportion, the degree of inaction and the energy thickness factor.

- In view of any round of operation in an organisation, we fairly determine the weight of every record depending on the coefficient of differentiation between entropy estimates.
- There are not many WSN use of DS proof hypothesis in hub credits while steering. PROPOSED ROUTING MODEL sees three records as evidence and determines if the hub should become the next hop as its goal of recognition. We use DS combination rules creatively to record the hub wire and take the results as the basis for guiding decisions.
- The re-launch results show that PROPOSED ROUTING MODEL is able to effectively decrease energy use as well as to drag out network life in energy-driven networks and also to achieve a reduced package disaster rate and increase information transmission trustworthiness.

3.3 NEIGHBOR DISCOVERY BLOCK

The Neighbor Discovery Block is enabled on network bootstrapping or by presenting each new hub. At this point, each hub sends and sends a one-spring message to get to know the next hubs. The message of welcome is accompanied by: {message type = 1 [one byte], sending ID [one byte], MAC [x bytes]} of the plot, if any, by confirming the MAC using the public key of the sender (hub ID) for PBC, or by the symmetrical key for ECC, as specified by the manager on the initialization block. Should the bundle not be correct, the recipient hub will rundown the sender in its neighbours and reply using a "unicast" which replies the bundle back to the sender to confirm its neighbourhood between them. The message of a welcome response has the same setup as the Message Type = 2 welcome message. Likewise all hubs transmit hello keep alive message to their neighbours, every fixed time, set and synchronised by BS. In the Initialization Block the executive sets the

time frame, and the time for submitting movement reports is taken several times. This welcome message is exactly the same as a welcome message with the Message Type = 3, and is used to ensure that there is a hub that keeps the BS refreshed with right data. The endless provision of a welcome, live message clearly reinforces without a response the situation with its neighbours. Note that if a confirmation plot set should occur, there should be labelled and checked a broad range of Hi messages for confidence in the characters of your neighbour.

3.4 BROADCASTING NODES

Another sort of attack is a preliminary one which triggers the organisation by communicating parcels to all its neighbours continuously. In two phases this kind is recognised and separated. Firstly, because the whole of their non-DL neighbours drop this parcel, the effect of the transmission is privately removed. As regards the fast transmission speed of the hub, the BS identifies that the hub will either wait or detach it immediately, depending on the time frame overhead of what it should be.

CHAPTER 4

ALGORITHM AND IMPLEMENTATION

4.1 ALGORITHMS

4.1.1 *ECC Algorithm (Elliptical Curve Cryptography)*

It is important to get basic climate information in military applications because such complex interaction helps information. The methods used to move the issue led to the suggested work helping to move the basic information safer in the multi-faceted WSN design. Encode the data using insignificant key size was done with the elliptic curve encoding part.

ECC Elliptic Curve Cryptography's essential idea is based on three factors

- The primary number to be selected as the largest
- A state of the curve
- Bending Public Point

Private Key: Private number selected

Key to the public: The public spotted Number of private opportunities In this type of context, the calculation of the private clay from the public key should be elliptical logarithmic work. Elliptic curvature depends on the focus that should meet the requirement. $y^2 = x^3 + \text{sleeve} + b$ Points may be marked with another point in the diagram. There are two areas where you can find an additional point together: $B = C A \text{ dab}$. $C = D A \text{ dab}$ $A D D = E$

Confidence estimate: The trust assessment of the neighbourhood nodes is presented in this section. The fundamental comprehension of the trust is summed up as:

Confidence is a connection created for a particular action between two entities. One individual relies in particular on the other to take an action. The first entity is referred to in this study as the subject, the second entity is the agent. Therefore, {subject: agent; action} is the notation used for describing trust relationship.

Allow $T\{subject: agent; action\}$ to indicate the trust value of the relationship of trust {subject: agent; action} and $P\{subject: agent; action\}$ to indicate the likelihood of the agent to behave in the subject's viewpoint. The theory of information notes that entropy is an insecurity measure; thus, entropy-based trust value as an error! $I\{T\{trust, agent, action\} H(p), for 0.5 \leq p \leq 1, H(p) = 1, for 0 \leq p \leq 0.5$ – reverse-reduce – reverse-release – reverse-reduce: $<(1)$ where, $H(p) = -$ reverse-reduce the reverse-release: $P = P [support: agent; actions]$ The confidence is an ongoing true number in this work $[-1, 1]$. The following characteristics are met by this description.

- If $P = 1$, the subject most trusts the agent and the value of the trust is 1
- If $P = 0$, the subject rejects most of the agent and the confidence value is -1
- The subject does not know the handler, the confidential value is 0 if $P = 0.5$ The trust value is generally negative for 0 to P

$$P(n) = \frac{1}{n(N)} k P(n) P(n) + = + \quad (4.1)$$

This calculates the likelihood of all the trials by the distribution below of the Bernoulli.

$$N K (N K) K P\{n(N)=K\} = p(1-p)^{-(N-K)} K \{n(N) = K\} = p(1-p)^{-(N-K)} \quad (4.2)$$

P = average chance of success of each packet transmission N = Total number of source packets; K = Number of neighbourhood node-transmitted packets

Secure road creation via antnet: if a node (source) wants to set a path to the other node (destination), the source will first try to find several destination routes. The source then attempts to identify the packet-forwarding credibility of the nodes on the routes from its own confidence record or by asking for recommendations. The source finally selects the reliable data transmission path. The source node will update the trust records following transmission on the basis of its track consistency observation. The following steps lead to the discovery of the route by the use of the Antnet algorithm:

- Every source starts several forward agents packets via multi-hop propagation to the destination. The route is randomly chosen based on the existing routing table
- Forward agent packets build a stack that pushes every visited node during travel times, trust values and traffic intensities as shown in the Figure. Some backward agent packets are sent to the source, when the packets get to the destination. The backward agent packets inherit the parameters of the stacks during this period. The parameter is popped and check again as shown in the figure. The packages for returning agents include the confidence values, the intensities of traffic and delays of the routes to the source. The optimum destination route will be selected from source[8-9]. Figure shows the hidden exchange of the source and the destination and the ECC. The total number of points in Elliptic curve, including the point on infinity. G Point on Elliptical curve with order ' n ' N The secret keys N_A, N_B The public clues P_A, P_B

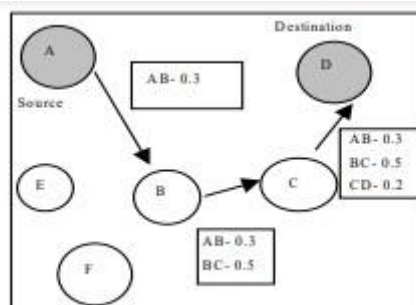


Fig. 4.1 Forward route discovery (ECC)

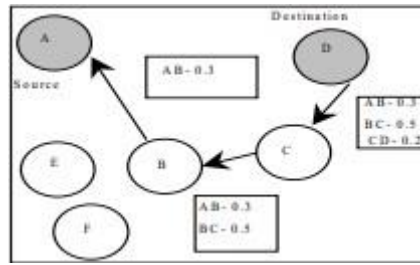


Fig. 4.2 Backward route discovery (ECC)

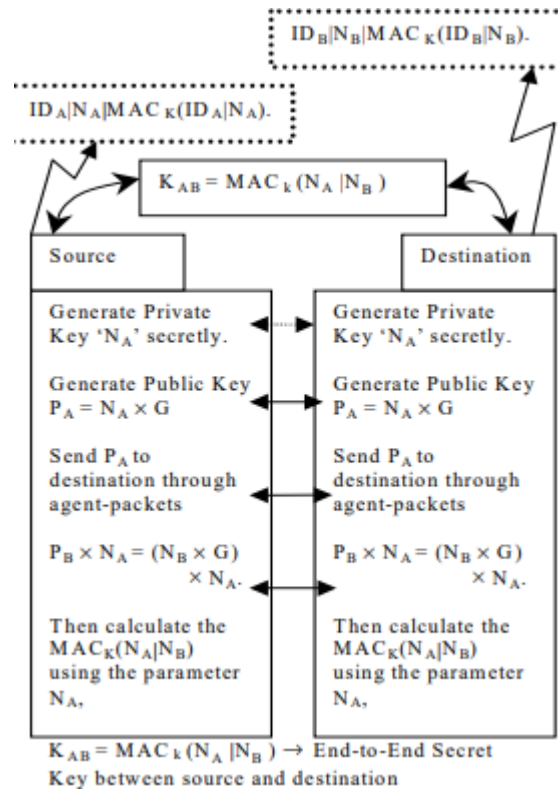


Fig. 4.3 Key generation in ECC

4.1.2 RSA Algorithm (Rivest Shamir Adleman)

RSA Algorithm In this paper, the data is encoded utilizing RSA public key calculation KEY GENERATION FOR RSA ALGORITHM

- It is needed to choose two indivisible numbers p and q haphazardly.
- $n = p * q$ and $\phi(n) = (p-1) * (q-1)$ be registered

- One discretionary whole number „e“ is to be chosen with $1 < e < \phi(n)$ to such an extent that $\gcd(e, \phi(n))=1$
- The whole number d is to be registered which fulfills $1 < d < \phi(n)$ and $ed \equiv 1 \pmod{\phi(n)}$
- Return (n,e,d) Public key parts are e and n where as Private key is d. The data is enciphered by utilizing public key of the recipient and translated by utilizing private key of the collector. RSA ENCRYPTION SCHEME Input: The plain content $m \in [0, n - 1]$ and The public key i.e (n, e) Output: Ciphared text c.

$c = m^e \pmod{n}$ must be processed

The encoded text, c must be returned.

RSA DECRYPTION SCHEME Input:

RSA private key d and the encoded text c Output: Plain content m.

$m = c^d \pmod{n}$ must be registered.

The unraveled text m must be returned which is the first content.

We evaluated the time a hub spends well above that needed for ordinary AODV for each overhead test. All experiments on the Zauruses were carried out with only the critical programming (i.e., no graphical login supervisor, no X worker, and so forth) In the SAODV tests, we calculate the age and SSE approval which needs an advanced mark/check and hash calculation. These tests have been conducted in MD5 and a 256-cycle RSA key pair has been used to carry out computerised signature and verification. For every estimate, there were 1000 tasks and 1000 estimates in general. The aftereffects of our SAODV experiments are shown in Table 1. In this way, the hub burns 30.8 milliseconds generating the SSA to send an RREQ, RREP, or HELLO message. The crucial impact on execution is when HELLO messages SSE are generated as they are often sent. In order to agree with AODV in particular, every HELLO INTERVAL millisecond should be sent a HELLO message, unless any messages have been exchanged during the past. This means that single RREQ and RERR messages will prevent the transmission of a

HELLO message, as every other message is unicast. This may, of course, put enormous weight on each hub.

Each RREQ and RREP receives a delay of 3.8 milliseconds at a jump that progresses SAODV because any message with an SSE is accepted prior to any further preparation. HELLO messages also set aside a related effort for approval. Although $\text{ALLOWED HELLO LOSS} * \text{HELLO INTERVAL}$ milliseconds should pass by until a link is decided and a neighbour should be removed from his director's table, the current status of those neighbours, whose HELLO parcels are deferred in consent, might be hesitant on a hub with a few neighbours and a large number of information to advance. We test the RREP-, HELLO- and R ACK messages by bundle overhead in TAODV. The broad system limits examined in [15] do not affect the TAODV overhead in any of the tests we have conducted. However, these qualities were necessary to render the RSV count. The following system has used wide-ranging considerations for all of the TAODV tests: $l = 0.8$, $p = 0.6$, $ph = 0.4$, $pc = 0.2$, $\alpha_1 = 0.4$, $\alpha_2 = 0.4$, and $\alpha_3 = 0.2$. The small task season saw an estimated 1,000,000 work carried out and an estimate of 5,000 was made. The results for the TAODV tests are shown in Table 2. As the results show, TAODV's overhead per plot is substantially less when compared to SAODV. The R ACK bundles are elaborated by the fundamental source of the overhead. As the R ACK bundles are new parcels rather than parcel increases, it was necessary to give a bundle of cradles each time a R ACK bundle was sent in the message, sending our implementation arrangement. The parcel cushion was designated at that point and the expansion was only composed into open space toward the end with various messages extended. This distinction led to the overhead of the R ACK communication of 0.193ms

4.2 IMPLEMENTATION

NS (version 2) is a tool for re-enacting the network open-source. It is a subject structured in C++ and Otol and is a discrete event based test system. NS is used in network studies for refocusing different forms of wired/distant areas and deep regional associations, network shows such as TCP and UDP, FTP, Telnet, CBR and VBR traffic sources, switch board frames such as Drop Tail, RED and

CQB, and monitoring computations like Dijkstra. In C++ and Otcl, Ns2 is written to detach the use of control and data. The test system supports a leadership class hierarchy in C++ and looks at movement within the Otcl interpreter (deciphered progression).

The inspiration for the use of two languages in ns2 is that various tasks vary. For instance, the restoration of the shows requires profitable bytes and group headers to monitor the runtime. Of course, in-network questioning where a few constraints are contrasted and an opportunity to change the model and run it again is more important in order to rapidly evaluate various circumstances.

In ns2, C++ is used to unmistakably use shows and when everything is said in order to achieve this, when every bundle of a stream has to be ready. For example, C++ is the language of choice, if you need to perform another coating discipline. Naturally, once the strategy and the plan are responsive. Otcl works slowly, but in general the progression of proliferations can be modified rapidly. Ns2 allows complete C++ items to be opened to the Otcl arbiter. Therefore, items from the Tcl level can be regulated from the moment C++.

4.2.1 OTCL Essentials

This section shows the phrases in the Otcl language and some basic orders used by ns2. Before moving to the part that manages to shape the real recovery, it is important to check how Otcl operates.

4.2.2 Assigning Esteems to Factors

In tcl, qualities can be reduced to factors and these qualities can be included in orders in addition:

Fix a set of 5 b [expr \$a/5] [Expr \$a/5]

The vector "5" is designated in the primary line. In the subsequent line, an incentive for variable b is used for the after-effect from the order [expr \$a/5],

which is raised to 1. (including 1). The "\$" symbol is used to obtain a value found in an order substitution variable and square parts are a sign.

4.2.3 *Procedures*

New systems can be characterised by procurement. The primary argument to be procured is the name of the tactic, and the following argument includes an overview of the names of the dispute. For example, the following can be characterised as a method which calculates two numbers:

```
Full proc {a b}
```

```
$a + $b Expr
```

```
}
```

The device below shows a number factory:

```
Factorial provision a {
```

```
{{ $a <= 1 } { on the off chance
```

```
Please return 1
```

```
}
```

```
#here again the device is named
```

```
[factorial [expr$x-1]] Express $x * [factorial]
```

```
}
```

It can also be conceived as a contentious list for an unfulfilled string. Nonetheless, the factors used by the strategy should be defined globally for this situation. Examples include:

```
Proc whole {}
```

```
A b all over the world
```

```
$a + $b Expr
```

```
}
```

4.2.4 *Files and Records*

For the order reading in tcl, a document can be opened:

Set the [open test.dat r] testfile

A overview with an order can be placed on the main line of the document:

Get the list of \$prooffile

It is currently possible for orders to be issued for the components of the rundown (component numbers start from 0):

First placed [lindex \$list 0] Set first

Set second [lindex \$list 1] Set second

A record may essentially be written with an order of suppression:

Sets the [open test.dat w] testfile

Puts "testi" on \$testfile

4.2.5 *Calling Subprocesses*

The Order Manager performs and tightly hangs for completion. Executive use means that the shell programme is supplied with an order line. To delete a record, for example:

```
rm $testfile executive
```

The executive order is particularly important if a tcl material from inside another tclscript has to be called up. For example, to run the example.tcl tcl-content on different occasions and to evaluate the "test" boundary from 1 to 10, you can type the accompanying tcl-contents in the following line:

```
{set ind 1} {$ind <= 10} {including ind} } {  
Put the $ind test  
Executive ns test.tcl  
}
```

4.2.6 *Creating the Geography*

To have the option to run a reproduction situation, an organization geography should initially be made. In ns2, the geography comprises of an assortment of hubs and connections.

Before the geography can be set up, another test system object should be made toward the start of the content with the order:

```
set ns [new Simulator]
```

The test system object has part works that empower making the hubs and the connections, interfacing specialists and so forth All these fundamental capacities can be found from the class Simulator. When utilizing capacities having a place with this class, the order starts with "\$ns", since ns was characterized to be a handle to the Simulator object.

4.2.7 *Nodes*

New hub articles can be made with the order:

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

```
set n3 [$ns node]
```

The part capacity of the Simulator class, called "hub" makes four hubs and appoints them to the handles n0, n1, n2 and n3. These handles can later be utilized when alluding to the hubs. In the event that the hub isn't a switch however an end framework, traffic specialists (TCP, UDP and so on) and traffic sources (FTP,CBR and so forth) should be set up, i.e, sources should be connected to the specialists and the specialists to the hubs, individually.

4.2.8 *Agents, applications and traffic sources*

The most widely recognized specialists utilized in ns2 are UDP and TCP specialists. If there should be an occurrence of a TCP specialist, a few kinds are accessible. The most widely recognized specialist types are:

- Agent/TCP - a Tahoe TCP sender
- Agent/TCP/Reno - a Reno TCP sender
- Agent/TCP/Sack1 - TCP with specific affirmation

The most well-known applications and traffic sources gave by ns2 are:

- Application/FTP - produces mass information that TCP will send
- Application/Traffic/CBR - produces bundles with a steady piece rate
- Application/Traffic/Exponential - during off-periods, no traffic is sent. During on-periods, bundles are produced with a consistent rate. The length of both on and off-periods is dramatically circulated.
- Application/Traffic/Trace - Traffic is produced from a follow record, where the sizes and interarrival seasons of the parcels are characterized.

Notwithstanding these instant applications, it is conceivable to produce traffic by utilizing the techniques given by the class Agent. For instance, on the off chance that one needs to send information over UDP, the technique send(int n bytes) can be utilized at the tcl-level gave that the udp-specialist is first designed and appended to some hub.

The following is a finished illustration of how to make a CBR traffic source utilizing UDP as transport convention and connect it to hub n0:

```
set udp0 [new Agent/UDP]
$ns connect specialist $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 append specialist $udp0
```

```
$cbr0 set packet_size_ 1000
```

```
$udp0 set packet_size_ 1000
```

```
$cbr0 set rate_ 1000000
```

A FTP application utilizing TCP as a vehicle convention can be made and appended to hub n1 similarly:

```
set tcp1 [new Agent/TCP]
```

```
$ns append specialist $n1 $tcp1
```

```
set ftp1 [new Application/FTP]
```

```
$ftp1 append specialist $tcp1
```

```
$tcp1 set packet_size_ 1000
```

The UDP and TCP classes are both kid classes of the class Agent. With the articulations [new Agent/TCP] and [new Agent/UDP] the properties of these classes can be consolidated to the new items udp0 and tcp1. These items are then joined to hubs n0 and n1. Then, the application is characterized and appended to the vehicle convention. At long last, the design boundaries of the traffic source are set. In the event of CBR, the traffic can be characterized by boundaries rate_ (or proportionally interval_, deciding the interarrival season of the parcels), packetSize_ and random_. With the random_ boundary it is conceivable to add some irregularity in the interarrival seasons of the bundles. The default esteem is 0, implying that no irregularity is added.

4.2.9 Traffic Sinks

In the event that the data streams are to be ended without handling, the UDP and TCP sources must be associated with traffic sinks. A TCP sink is characterized in the class Agent/TCP Sink and a UDP sink is characterized in the class Agent/Null.

A UDP sink can be appended to n2 and associated with udp0 in the accompanying manner:

```
set invalid [new Agent/Null]
```

```
$ns join specialist $n2 $null
```

```
$ns associate $udp0 $null
```

A standard TCP sink that makes one affirmation for every a got parcel can be appended to n3 and associated with tcp1 with the orders:

```
set sink [new Agent/Sink]
```

```
$ns join specialist $n3 $sink
```

```
$ns associate $tcp1 $sink
```

There is likewise a more limited approach to characterize associations between a source and the objective with the order:

```
$ns make association <srctype> <src> <dsttype> <dst> <pktclass>
```

For instance, to make a standard TCP association somewhere in the range of n1 and n3 with a class ID of 1:

```
$ns make association TCP $n1 TCPSink $n3 1
```

One can undoubtedly make a few TCP-associations by utilizing this order inside a for-circle.

4.2.10 *Links*

Connections are needed to finish the geography. In ns2, the yield line of a hub is executed as a component of the connection, so when making joins the client likewise needs to characterize the line type.

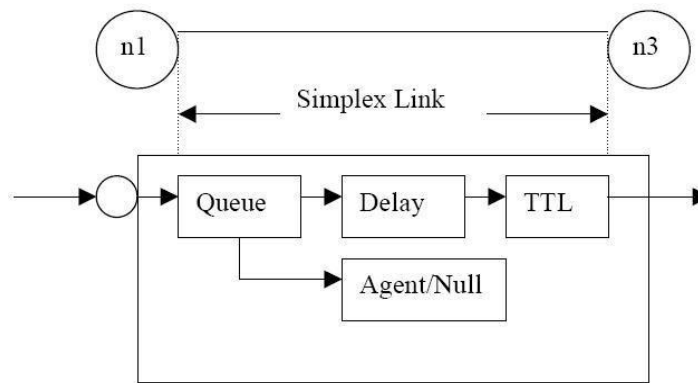


Fig. 4.4 Link in NS2

This Figure shows the development of a simplex connection in ns2. On the off chance that a duplex-interface is made, two simplex connections will be made, one for every course. In the connection, bundle is first enqueued at the line. After this, it is either dropped, passed to the Null Agent and liberated there, or dequeued and passed to the Delay object which mimics the connection delay. At long last, the TTL (time to live) esteem is determined and refreshed.

Connections can be made with the accompanying order:

```
$ns duplex/simplex-connect endpoint1 endpoint2 data transfer capacity
defer line type
```

For instance, to make a duplex-connect with DropTail line the executives somewhere in the range of n0 and n2:

```
$ns duplex-interface $n0 $n2 15Mb 10ms DropTail
```

Making a simplex-interface with RED line the executives somewhere in the range of n1 and n3:

```
$ns simplex-interface $n1 $n3 10Mb 5ms RED
```

The qualities for data transmission can be given as an unadulterated number or by utilizing qualifiers k (kilo), M (super), b (cycle) and B (byte). The postponement can likewise be communicated in a similar way, by utilizing m (milli)

and u (micro) as qualifiers. There are a few line the board calculations actualized in ns2, yet in this activity just Drop Tail and RED will be required.

4.2.11 *Tracing and Checking*

To have the option to compute the outcomes from the reenactments, the information must be gathered by one way or another. NS2 upholds two essential checking abilities: follows and screens. The follows empower recording of parcels at whatever point an occasion, for example, bundle drop or appearance happens in a line or a connection. The screens give a way to gathering amounts, for example, number of parcel drops or number of showed up bundles in the line. The screen can be utilized to gather these amounts for all bundles or only for a predetermined stream (a stream screen).

4.2.12 *Traces*

All occasions from the reenactment can be recorded to a document with the accompanying orders:

```
set trace_all [open all.dat w]
$ns follow all $trace_all
$ns flush-follow
close $trace_all
```

To start with, the yield document is opened and a handle is connected to it. At that point the occasions are recorded to the document determined by the handle. At long last, toward the finish of the recreation the follow support must be flushed and the record must be shut. This is typically finished with a different completion system. On the off chance that connections are made after these orders, extra articles for following (EnqT, DeqT, DrpT and RecvT) will be embedded into them.

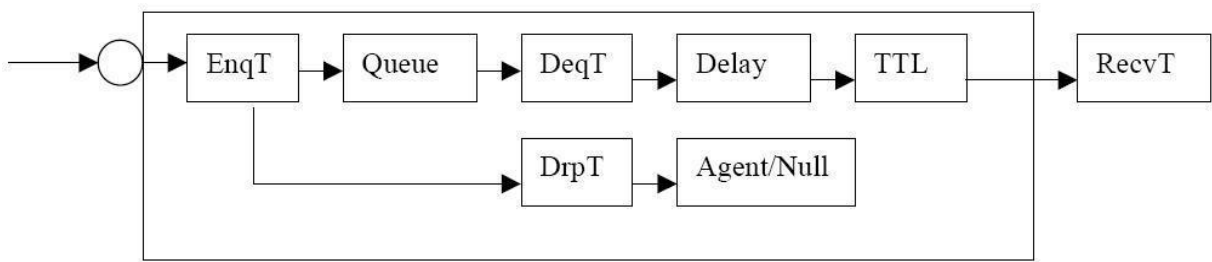


Fig. 4.5 Link in NS2 when tracing is enabled

New items will at that point keep in touch with a follow record at whatever point they get a bundle. The configuration of the follow document is following:

```

+ 1.84375 0 2 cbr 210 - 0 0.0 3.1 225 610
- 1.84375 0 2 cbr 210 - 0 0.0 3.1 225 610
r 1.84471 2 1 cbr 210 - 1 3.0 1.0 195 600
r 1.84566 2 0 ack 40 - 2 3.2 0.1 82 602
+ 1.84566 0 2 tcp 1000 - 2 0.1 3.2 102 611
- 1.84566 0 2 tcp 1000 - 2 0.1 3.2 102 611

+ : enqueue
- : dequeue
d : drop
r : get
  
```

The fields in the follow document are: kind of the occasion, recreation time when the occasion happened, source and objective hubs, bundle type (convention, activity or traffic source), parcel size, banners, stream id, source and objective locations, grouping number and bundle id. Notwithstanding following all occasions of the reproduction, it is additionally conceivable to make a follow object between a specific source and an objective with the order:

\$ns make follow <type> <file> <src> <dest>

where the sort can be, for example,

- Enque - a parcel appearance (for example at a line)

- Deque - a bundle flight (for example at a line)
- Drop - bundle drop
- Recv - bundle get at the objective

Following all occasions from a reenactment to a particular document and afterward figuring the ideal amounts from this record for example by utilizing perl or awk and Matlab is a simple way and appropriate when the geography is moderately straightforward and the quantity of sources is restricted. In any case, with complex geographies and numerous sources this method of gathering information can turn out to be excessively lethargic. The follow records will likewise burn-through a lot of circle space.

4.2.13 *Monitors*

With a line screen it is conceivable to follow the insights of appearances, flights and drops in either bytes or bundles. Alternatively the line screen can likewise keep a fundamental of the line size over the long haul.

For example, if the hubs n0 and n1 are connected, the line screen can be set as follows:

```
set qmon0 [$ns screen line $n0 $n1]
```

The bundle appearances and byte drops can be followed the orders:

```
set parr [$qmon0 set parrivals_]
```

```
set bdrop [$qmon0 set bdrops_]
```

4.2.14 *Controlling the recreation*

After the geography of reproduction, specialists are created, and so on, recreation begins and ends and various occasions must be reserved. The reconstruction can be started and stopped by order

```
$ns at $simtime "finish"
```

```
$ns run
```

The main order expects to complete the process towards the completion of the reenactment, and then the recreation begins. The completion plan must be designed to flush out follow-up assistance, close the documents and finish off the programme. Alternatively, it will begin NAM, post-cycle data and plot this data. This is also possible.

The completion method needs to contain at any rate the accompanying components:

```
proc finish {} {  
    worldwide ns trace_all  
  
    $ns flush-follow  
  
    close $trace_all  
  
    leave 0  
}
```

Different occasions, for example, the beginning or halting occasions of the customers can be planned for the accompanying way:

```
$ns at 0.0 "cbr0 start"  
  
$ns at 50.0 "ftplstart"  
  
$ns at $simtime "cbr0 stop"  
  
$ns at $simtime "ftpl stop"
```

If you have characterized your own programs, you may also prepare to start the process regularly, for instance, in the following way:

4.3 SIMPLE STIMULATION EXAMPLE

This section offers a simple NS reproduction and clarifies what each line does. Model 3 is the OTcl material for the basic organizational arrangement and for Figure 4 to reenact. In Example 3 of both of those records "ns-simple.tcl" and "nS-simple.tcl" in the short shell to perform this reconstruction, repeat the code

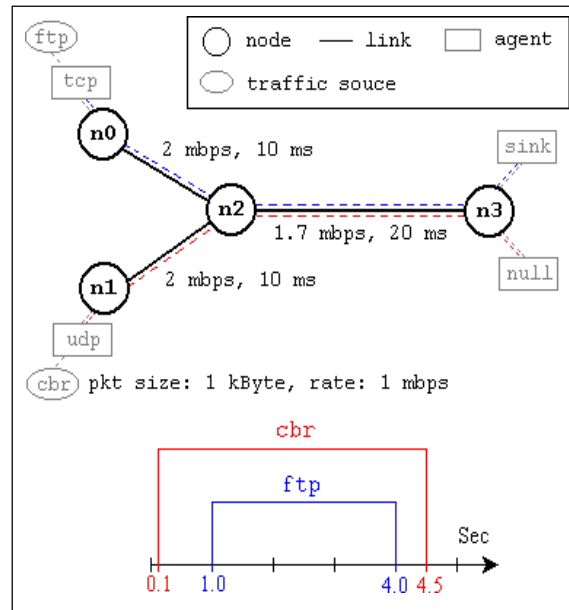


Fig. 4.6 A Simple Network Topology and Stimulation Scenario

There are 4 nodes (n0, n1, n2, n3) on the network as indicated above. There are two bandwidth Mbps and 10 ms of late duplex connections between n0 and n2 and n1 and n2. The bandwidth and 20 ms of delay are 1.7 Mbps of the duplex between n2 and n3. The maximum size for each node is 10 DropTail queues. A "tcp" agent is attached to n0, and the tcp "sink" agent is connected to n3. The parameters that a tcp agent will create are 1KByte by default. A tcp "sink" agent generates the received ACK packets and sends them to the sender (tcp agent) and frees them. A "udp" agent attached to n1 is bound to n3 by a "empty" agent. A 'zero' agent just frees the obtained packets. A traffic generator "ftp" and a "cbr" are connected to the "tcp" and the "udp" agents, and a "cbr" is configured to produce 1 KByte of 1 Mbps packets. The "cbr" starts at 0.1 sec and stops at 4.5 sec and the "ftp" starts at 1.0 sec and stops at 4.0 seconds.

```

#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns nam trace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]

```

```

$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation

```

Example. A Simple NS Simulation Script

Coming up next is the clarification of the content above. By and large, a NS content beginnings with making a Simulator object example.

set ns [new Simulator]: creates a NS test system object occasion, and appoints it to variable ns (*italics is utilized for factors and qualities in this segment*). What this line does is the accompanying:

- Create a scheduler (default is schedule scheduler)

- The "Test system" object has part works that do the accompanying:
 - Create compound items like hubs and connections (depicted later)
 - Connect network part protests made (ex. join specialist)
 - Set network segment boundaries (generally for compound articles)
 - Create associations between specialists (ex. make association between a "tcp" and "sink")
 - Specify NAM show choices
- Etc.

A large portion of part works are for reenactment arrangement and booking, anyway some of them are for the NAM show. The "Test system" object part work usage are situated in the "ns-2/tcl/lib/ns-lib.tcl" document's color fid color: is to set color of the packets for a flow specified by the flow id (fid). This is for the NAM display, and has no effect on the actual simulation.

- \$ns namtrace-all document descriptor: It advises the test system to record reenactment follows in NAM input design. It additionally gives the document name that the follow will be kept in touch with later by the order \$ns flush-follow. Likewise, the part work follow everything is for recording the recreation follow in an overall configuration.
- proc finish {}: is called after this reproduction is more than by the order \$ns at 5.0 "finish". In this capacity, post-reenactment measures are indicated.
- set n0 [\$ns node]: The part work hub makes a hub.
- \$ns duplex interface node 1 node 2 defer line style transmission capacity: providing a couple of easy connections of the speed and delay of the transfer and combining these two hubs. In NS, the yield lines of a hub are performed as a link, so customers can decide the

type of lines during connections. DropTail line is used for the above reproduction material.

- `$ns node1 node2 line limits number`: this line sets the distance between node 1 and Node2 and the number indicated for all single-line connections.
- `Operation node1 $ns duplex-connect`: several lines are used for showing the NAM. This code line is used to screen the link line between node 1 and node 2 hubs. The customer can see the parcels on the line and the customer can eventually also feel the drop.
- The following activities are conducted in accordance with the fundamental organizational structure to coordinate traffic experts such as TCP and UDP, traffic sources such as FTP and CBR as well as to individual hubs and specialists.
- `Set tcp (new Agent/TCP)`: This line becomes a TCP expert. A client can name these elements in order to create traffic specialists or sources (Agent/TCP, Agnet/TCPSink, Application/FTP, etc.).
- `$ns enter specialist platform specialist`: a specialist object built on a hub is linked to a specialist object. This ability in fact refers to the additional part of the indicated center, which links the specialist provided to itself. So, by connecting `$n0 $tcp`, for example, a client may do something very similar.
- `$ns interface agent1 agent2`: The following is to establish a coherent organizational partnership between two experts who can speak with one another.
- Accepting that all the organization design is done, the following activity is compose a recreation situation (for example reenactment planning). The Simulator object has many booking part works.

Notwithstanding, the one that is generally utilized is the accompanying:

- \$ns at time "string": This part capacity of a Simulator object makes the scheduler (scheduler_ is the variable that focuses the scheduler object made by [new Scheduler] order toward the start of the content) to plan the execution of the predefined string at given recreation time. For instance, \$ns at 0.1 "\$cbr start" will settle on the scheduler decision a beginning part capacity of the CBR traffic source object, what begins the CBR to communicate information.
- After all organization setups, booking and post-reenactment system determinations are done, the solitary thing left is to run the recreation. This is finished by \$ns run.

4.4 EXTENDING NS: CREATING ANOTHER AGENT

One can broaden NS by adding new conventions. This segment will examine about how another specialist can be made in NS utilizing a model. The code in this segment executes a type of straightforward 'ping' convention. One hub will send n client characterized number bundles, each in turn at standard stretches, to another hub which will restore the parcel right away. For every bundle the sender will at that point ascertain the full circle time.

4.5 THE HEADER RECORD

First we need to make a header document "myPing.h" in which we will characterize the information structure for the myPing parcel header.

```

struct hdr_myPing {
    char ret;

    double send_time;

    // Header access methods

    static int offset_; // required by PacketHeaderManager

    inline static int& offset() { return offset_; }

    inline static hdr_myPing* access(const Packet* p) {
        return (hdr_myPing*) p->access(offset_);
    }
};

```

When the singe "ret" is on the way from the sender to the ping pit, it is set to '0' and on its way out, the singe "ret" is set to "1." The double time 'sends' is a time stamp determined to be delivered to the bundle. The sender then uses this time to determine the complete circle time. The rest of the software codes in the header are used for any parcel connection to the bundle header.

This code is a subset of the class 'My Ping Agent' as a 'Specialist' underclass.

```

class MyPingAgent : public Agent {
public:
    MyPingAgent();

    int command(int argc, const char*const* argv);

    void recv(Packet*, Handler*);

    void send_packets();

protected:
    int num_packets_;

    double wait_interval_;

    MyPingAgentTimer timer_;

    int count;
};

```

The num parcels vector contains the entire number of packages to be transmitted by the transmitter and the time between two progressive packets is "stand-by-stretch." The 'tally' vector maintains packet inspection to date. We must also characterize a clock controller (My ping agent timer) for the transmitter so that it can reprogram every time the bundle is sent. My Ping Agent uses a My Ping Agent Timer variable 'clock' to track time. The source code for the "myPing.h" header record is given below:

```
/* myPing.h */
#ifndef ns_myPing_h
#define ns_myPing_h
#include "agent.h"
#include "tclcl.h"
#include "packet.h"
#include "address.h"
#include "ip.h"
class MyPingAgent;
struct hdr_myPing {
    char ret;
    double send_time;

    // Header access methods
    static int offset_; // required by PacketHeaderManager
    inline static int& offset() { return offset_; }
    inline static hdr_myPing* access(const Packet* p) {
        return (hdr_myPing*) p->access(offset_);
    }
};
class MyPingAgentTimer : public TimerHandler {
public:
    MyPingAgentTimer(MyPingAgent *t) : TimerHandler() { t_ = t; }

protected:
    virtual void expire(Event *e);
    MyPingAgent *t_;
class MyPingAgent : public Agent {
public:
    MyPingAgent();
    int command(int argc, const char*const* argv);
    void rcv(Packet*, Handler*);
    void send_packets();
protected:
    int num_packets_;
    double wait_interval_;
    MyPingAgentTimer timer_;
    int count;
};
#endif
```

4.5.1 *Table of MyPing.h*

The C++ code for new Agent

First the linkage between the C++ code and Tcl code must be characterized.

```
static class kangAbHeaderClass : public PacketHeaderClass {
    public:
        kangAbHeaderClass() : PacketHeaderClass
            ("PacketHeader/MyPing", sizeof(hdr_myPing)) {
            bind_offset(&hdr_myPing::offset_);
        }
} class_kangAbhdrSnd;

static class MyPingAgentClass : public TclClass {
    public:
        MyPingAgentClass() : TclClass("Agent/MyPing") {}
        TclObject* create(int, const char*const*) {
            return (new MyPingAgent());
        }
} class_myPing;
```

The following piece of code is the constructor for the class 'MyPingAgent'. It ties the factors which must be gotten to both in Tcl and C++.

```
MyPingAgent::MyPingAgent() : Agent(PT_MYPING), timer_(this) , count(0)
{
    bind("packetSize_", &size_);
    bind("num_packets_", &num_packets_);
    bind("wait_interval_" &wait_interval_);
}
```

When an order Tcl for the class 'My Ping Agent' is executed, the capability 'order()' is taken into account. This will be "sender \$" for our situation (that is expected to be "sender" of the agent/mY ping class), since we have to send ping plots from the agent to another ping specialist. In the 'order()' work, we need to parse the order. If the order matches the order, the "send () packets" work will be done, which in turn sends each bundle and reprogramms to send the next parcel after the stand. If no match is found, it is transferred into the basic class capability of the 'order()' (the 'agent::command()') for this situation) with its contentions.

```

int MyPingAgent::command(int argc, const char*const* argv)
{
    if (argc == 2) {
        if (strcmp(argv[1], "send") == 0) {
            send_packets();
            // return TCL_OK, so the calling function knows that
            // the command has been processed
            return (TCL_OK);
        }
    }
    return (Agent::command(argc, argv));
}

void MyPingAgent::send_packets()
{
    if(count++ < num_packets_)
    {
        Packet *pkt = allocpkt();
        hdr_myPing *hdr = hdr_myPing::access(pkt);
        // Set the 'ret' field to 0, so the receiving node knows
        // that it has to generate an echo packet
        hdr->ret = 0;
        // Store the current time in the 'send_time' field
        hdr->send_time = Scheduler::instance().clock();
        // Send the packet
        send(pkt, 0);
    }
    if(count < num_packets_)
        timer_.resched(wait_interval_);
}

```

The 'recv()' function determines what measures to take when you receive a packet. If "ret" is 0, the packet has to be returned with the same value for the field "send time," but the field "ret" set to 1. If 'ret' is 1, it is measured and printed for the roundtrip time of the packet. The full "myPing.cc" source code is given as follows:

```

/* myPing.cc */

#include "myPing.h"

int hdr_myPing::offset_;

static class kangAbHeaderClass : public PacketHeaderClass {
public:
    kangAbHeaderClass() : PacketHeaderClass("PacketHeader/MyPing",
                                             sizeof(hdr_myPing)) {
        bind_offset(&hdr_myPing::offset_);
    }
} class_kangAbhdrSnd;

static class MyPingAgentClass : public TclClass {
public:
    MyPingAgentClass() : TclClass("Agent/MyPing") {}
    TclObject* create(int, const char*const*) {
        return (new MyPingAgent());
    }
} class_myPing;

/* constructor*/
MyPingAgent::MyPingAgent() : Agent(PT_MYPING), timer_(this), count(0)
{
    bind("packetSize_", &size_);
    bind("num_packets_", &num_packets_);
    bind("wait_interval_", &wait_interval_);
}

/* if timer expires just do a timeout */
void MyPingAgentTimer::expire(Event*)
{
    t_ -> send_packets();
}

int MyPingAgent::command(int argc, const char*const* argv)
{
    if (argc == 2) {
        if (strcmp(argv[1], "send") == 0) {
            send_packets();
            // return TCL_OK, so the calling function knows that
            // the command has been processed
            return (TCL_OK);
        }
    }
}

```

```

void MyPingAgent::send_packets()
{
    if(count++ < num_packets_)
    {
        Packet *pkt = allocpkt();
        hdr_myPing *hdr = hdr_myPing::access(pkt);
        // Set the 'ret' field to 0, so the receiving node knows
        // that it has to generate an echo packet
        hdr->ret = 0;
        // Store the current time in the 'send_time' field
        hdr->send_time = Scheduler::instance().clock();
        // Send the packet
        send(pkt, 0);
    }
    if(count < num_packets_)
        timer_resched(wait_interval_);
}

void MyPingAgent::rcv(Packet* pkt, Handler*)
{
    hdr_myPing *hdr=hdr_myPing::access(pkt);
    if (hdr->ret == 0) {
        // Send an 'echo'. First save the old packet's send_time
        double stime = hdr->send_time;
        // Discard the packet
        Packet::free(pkt);
        // Create a new packet
        Packet* pktret = allocpkt();
        // Access the Ping header for the new packet:
        hdr_myPing* hdrret = hdr_myPing::access(pktret);
        // Set the 'ret' field to 1
        hdrret->ret = 1;
        // Set the send_time field to the correct value
        hdrret->send_time = stime;
        // Send the packet
        send(pktret, 0);
    }
    else {
        // Get the receiving time of the packet
        double rcv_time = Scheduler::instance().clock();
        printf("RTT = %lf\n",rcv_time-hdr->send_time);
        // Discard the packet
        Packet::free(pkt);
    }
}

```

4.5.2 Table of myPing.cc

Presently save these two records "myPing.h" and "myPing.cc" in the "... ./ns-allinone-2.31/ns-2.31/" envelope. For great practice make a different index for all projects for client characterized conventions. In our model we have kept these two records in "... ./ns-allinone-2.31/ns-2.31/bwtools/MyPing" registry.

4.6 NECESSARY CHANGES IN NS

Subsequent stage is to adjust a portion of the NS framework records so the recently composed convention or specialist can work with the test system. The records that are should have been adjusted are appeared inside red air pockets

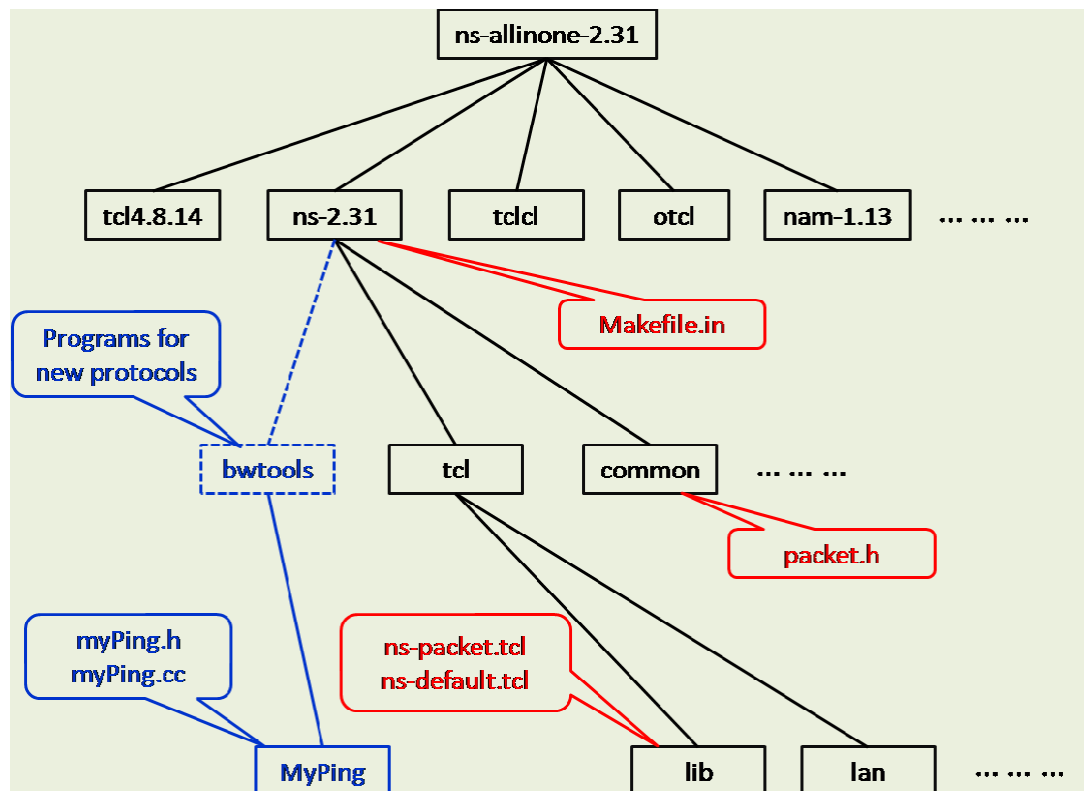


Fig. 4.7 NS directory structure

CHAPTER 5

RESULTS AND DISCUSSION

5.1 EDIT PACKET.H FILE

We need another parcel type for our MyPing specialist, in this way the initial step is to adjust the "\$NS/normal/packet.h" record (expecting that the \$NS climate variable focuses the/ns-allinone-2.31/ns-2.31 area). There you can discover the definitions for the bundle convention IDs (for example PT_TCP, PT_TELNET, and so forth) Add another definition for PT_PING there. In my altered variant of packet.h, the last couple of lines of enum packet_t{} resembles the accompanying code (it may look somewhat changed in changed NS discharges).

```
enum packet_t {
    PT_TCP,
    PT_UDP,
    .....
    // insert new packet types here
    PT_TFRC,
    PT_TFRC_ACK,
    PT_MYPING, // packet protocol ID for MyPing agent
    PT_NTTYPE // This MUST be the LAST one
};
```

You also have to edit the p_info() in the same file to include "MyPing".

```

class p_info {
public:
    p_info() {
        name_[PT_TCP]= "tcp";
        name_[PT_UDP]= "udp";
        .....
        name_[PT_TFRC]= "tcpFriend";
        name_[PT_TFRC_ACK]= "tcpFriendCtl";

        name_[PT_MYPING]="MyPing";

        name_[PT_NTTYPE]= "undefined";
    }
    .....
}

```

5.2 EDIT NS-PACKET.TCL DOCUMENT

Presently in "\$NS/tcl/lib/ns-packet.tcl" record look for the part for each prot { .. }. Inside this part add a passage for MyPing. It will look like after:

```

foreach prot {
# Common:
    Common
    Flags
    IP      # IP
    ... ..
    Pushback      # Pushback, router-to-router
    MyPing
# Multicast:
    aSRM  # mcast/srm.cc
    ... ..
# Other:
    Encap  # common/encap.cc
    IPinIP  # IP encapsulation
    HDLC    # High Level Data Link Control
}{
    add-packet-header $prot
}

```

```

foreach prot {
# Common:
    Common
    Flags
    IP      # IP
    ... ..
    Pushback      # Pushback, router-to-router
    MyPing
# Multicast:
    aSRM  # mcast/srm.cc
    ... ..
# Other:
    Encap  # common/encap.cc
    IPinIP  # IP encapsulation
    HDLC    # High Level Data Link Control
}{
    add-packet-header $prot
}

```

5.3 EDIT NS-DEFAULT.TCL

The default file for Tcl objects "\$NS/tcl/lib/ns-default.tcl" is the default values for Tcl objects. Please insert the next row to specify packet size, packet number and Agent/MyPing waiting interval.

```

Agent/MyPing set packetSize_ 200
Agent/MyPing set num_packets_ 3
Agent/MyPing set wait_interval_ .001

```

5.4 EDIT MAKEFILE.IN

The most recent change is the modification of "\$NS/Makefile.in." You must add "myPing.o" to the object file lists for ns. My version shows the final lines of the edited list:

```

OBJ_CC = \
    tools/random.o tools/rng.o tools/ranvar.o common/misc.o common/timer-
handler.o \
    common/scheduler.o common/object.o common/packet.o \
    ... ..
    bwtools/MyPing/myPing.o \
    tools/trafgen.o trace/traffictrace.o tools/pareto.o \

```

5.5 RECOMPILE NS

We must recompile the NS simulator after changes are made to the NS files. Use the following command for this purpose:

```
droy@aggarp3:~$ cd $NS
droy@aggarp3:ns-2.31$ pwd
/usr/share/ns-allinone-2.31/ns-2.31
droy@aggarp3:ns-2.31$ sudo ./configure
droy@aggarp3:ns-2.31$ sudo make
```

5.6 Simple example using MyPing agent

After recompiling the NS simulator we can use our MyPing agent in our simulations. Following code is simple tcl program that uses the MyPing agent.

```
set ns [new Simulator]
set nf [open out.tr w]
$ns trace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the trace file
    close $nf
    #Execute nam on the trace file
    exit 0
}
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 10Mb 50ms DropTail

#####Sender Side Stuff#####
set sender [new Agent/MyPing]
$ns attach-agent $n0 $sender
$sender set packetSize_ 1500
$sender set num_packets_ 5
$sender set wait_interval_ .05

#####Receiver Side Stuff
set recv [new Agent/MyPing]
$ns attach-agent $n1 $recv

#####Connect the sender with receiver
$ns connect $sender $recv

#Schedule events for the CBR agent
$ns at 0.0 "$sender send"
$ns at 10.0 "finish"
$ns run
```

5.7 RSA – NAM RESULTS

Algorithm : RSA (Rivest Shamir Adleman)
Input data : hello-2021
Transmitter node : 5
Receiver node : 0
Total nodes : 41

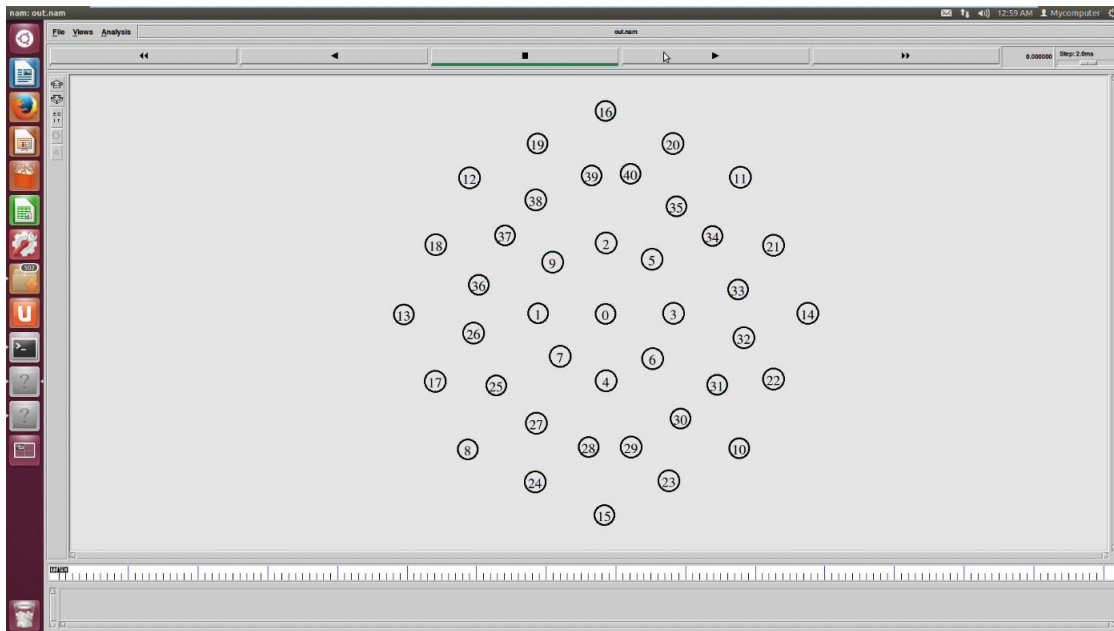


Fig. 5.1 Network Representation NAM output file (RSA)

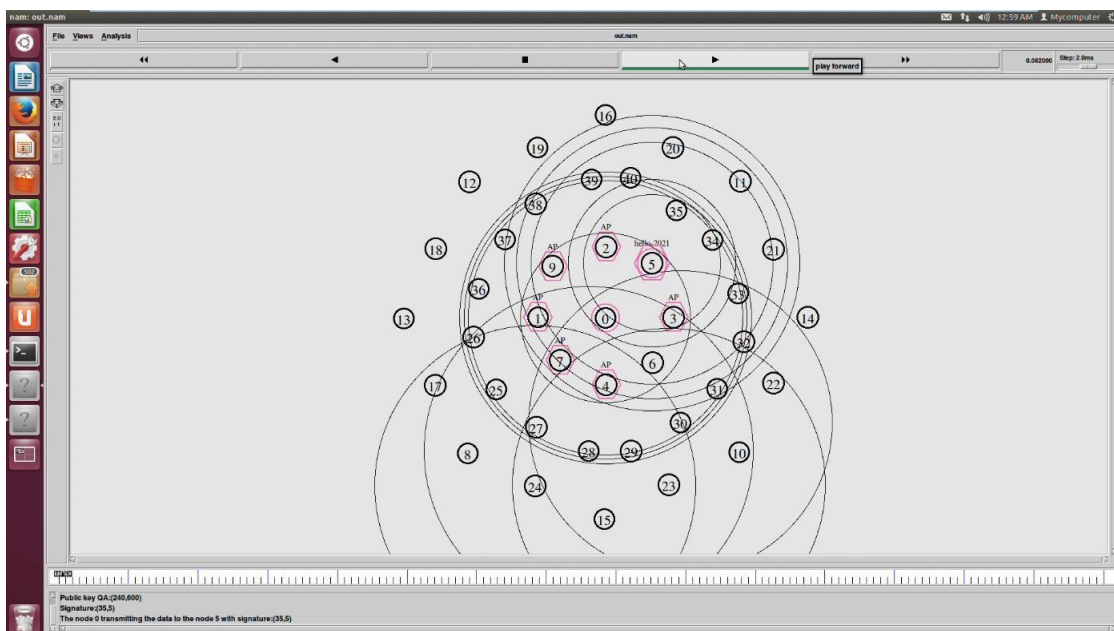


Fig. 5.2 Authentication between Nodes

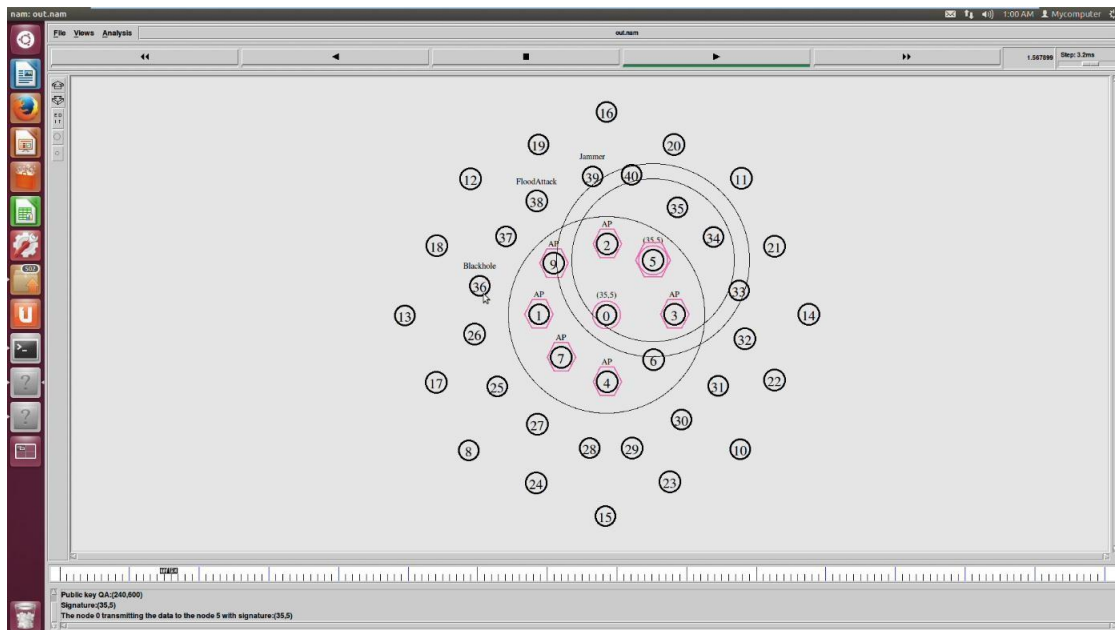


Fig. 5.3 Validing signature (keys) in Network

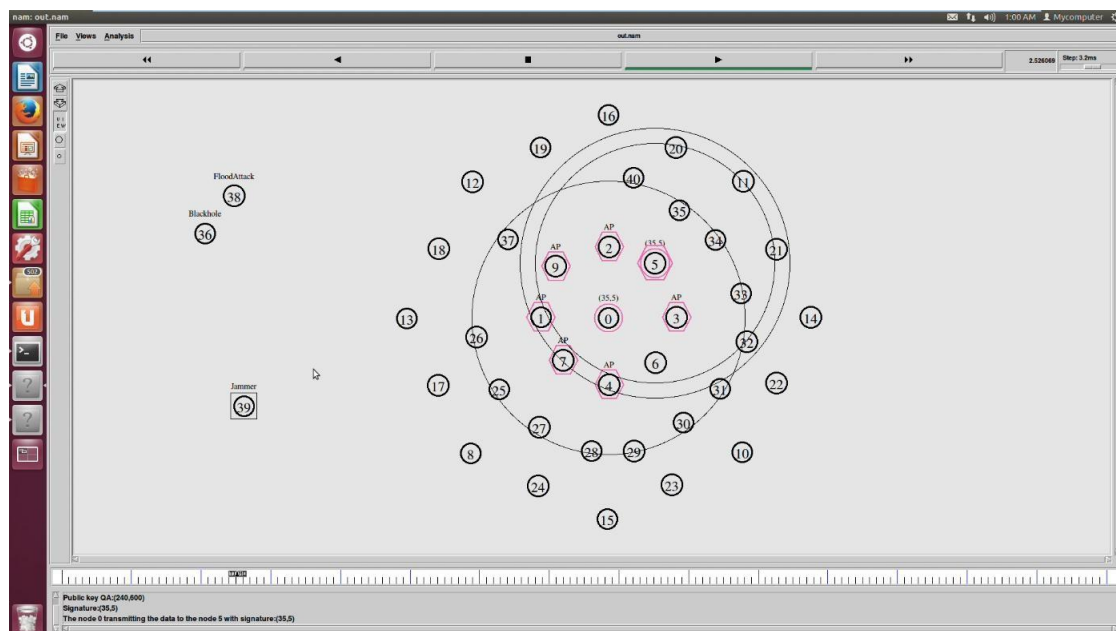


Fig. 5.4 Routing in Network over off the attackers

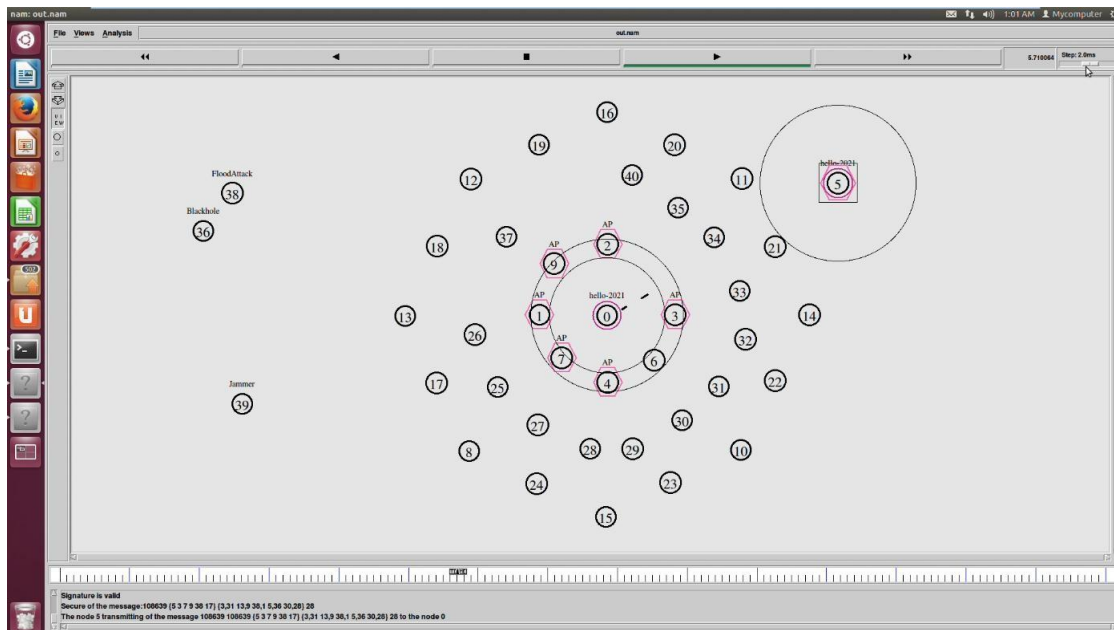


Fig. 5.5 Receiving data at Receiver node from transmitter node

5.8 ECC – NAM RESULTS

Algorithm : ECC (Elliptical Curve Cryptography)
 Input data : hello-2021
 Transmitter Node : 5
 Receiver Node : 1
 Total Nodes : 41

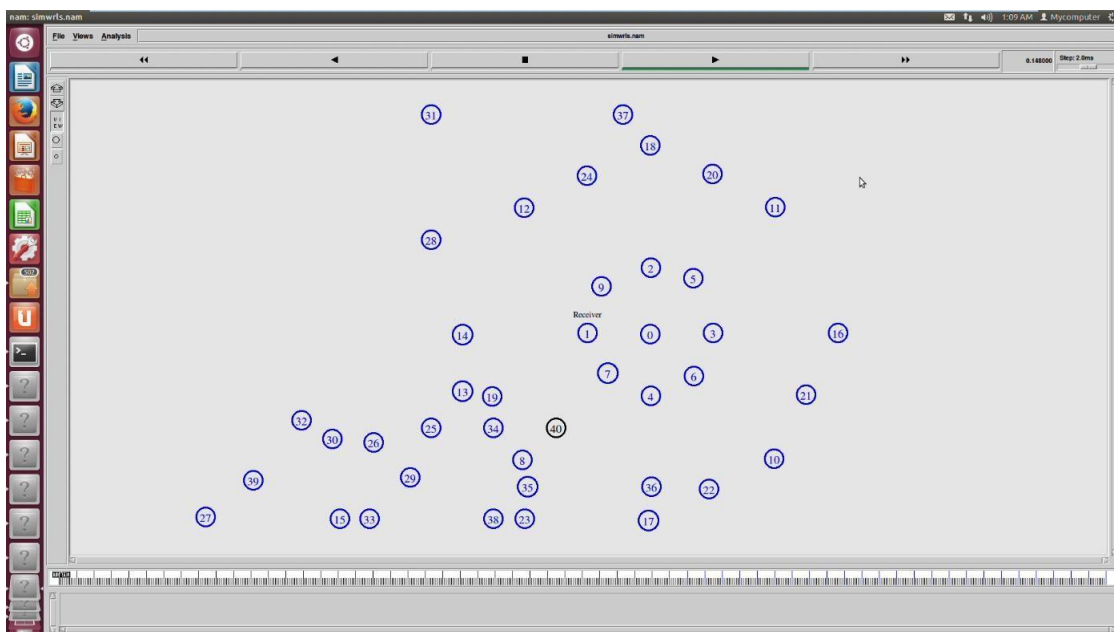


Fig. 5.6 Network Representation NAM output (ECC)

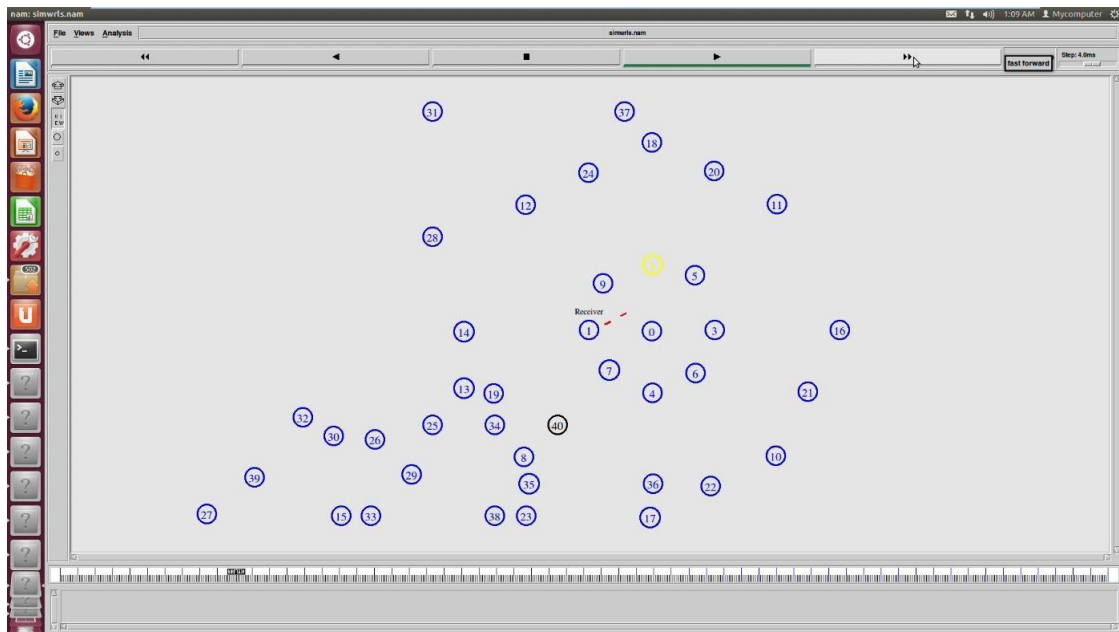


Fig. 5.7 Data transmission between Transmitter and Receiver nodes

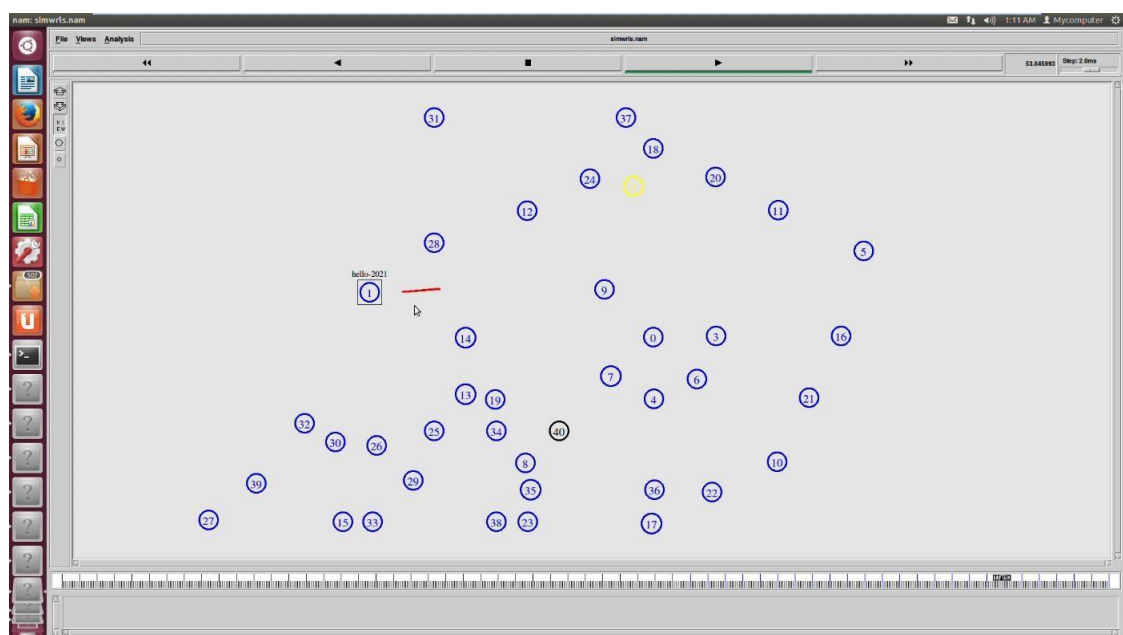


Fig. 5.8 Routing path between Source and Destination

5.8 PARAMETER ANALYSIS

5.8.1 RSA Results

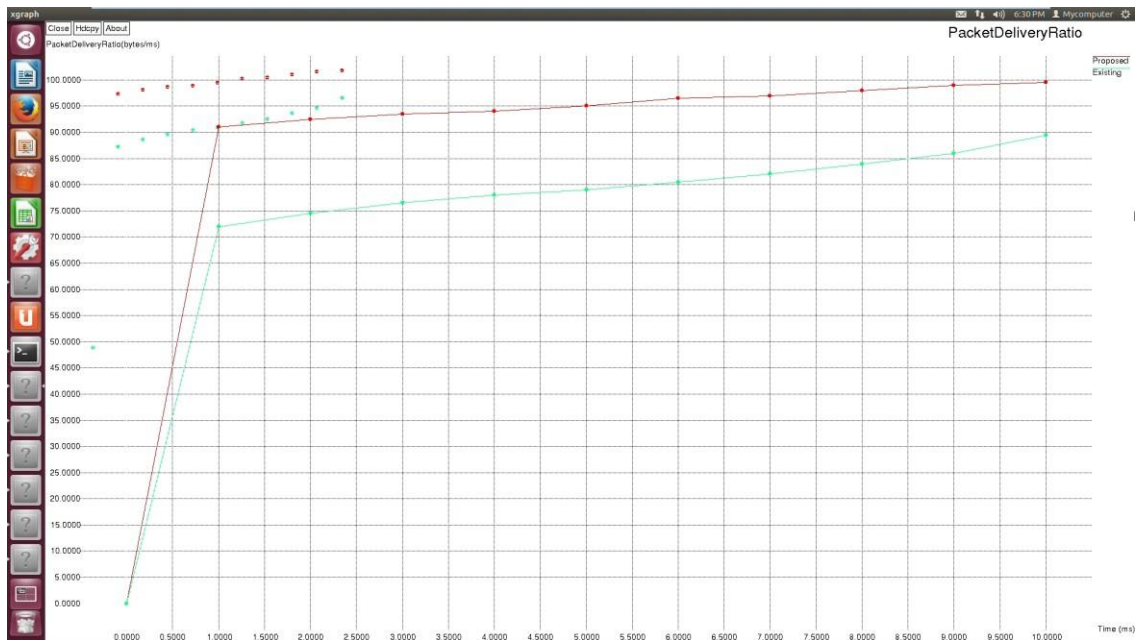


Fig.5.9 Packet Delivery Ratio

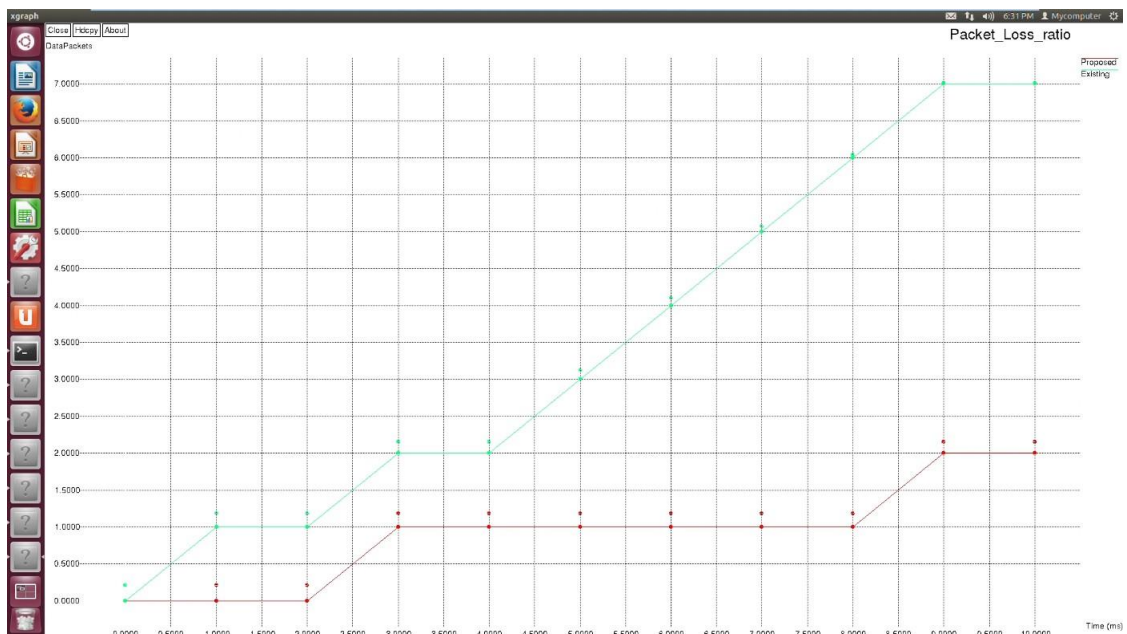


Fig. 5.10 Packet Loss Ratio

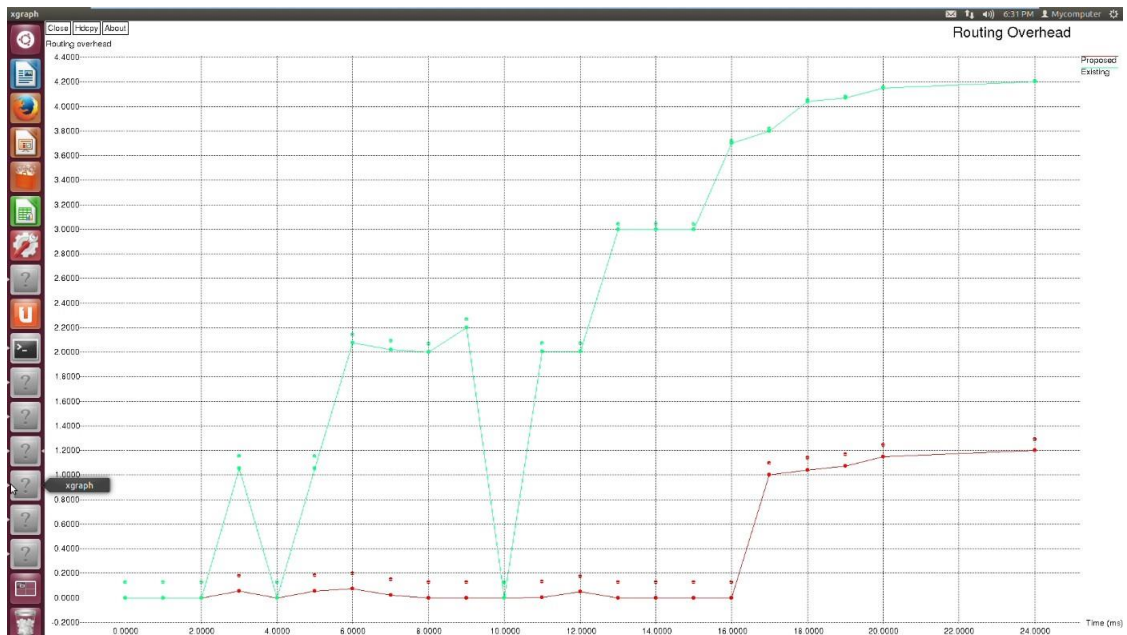


Fig. 5.11 Routing Overhead



Fig. 5.12 Comparison Of Energy

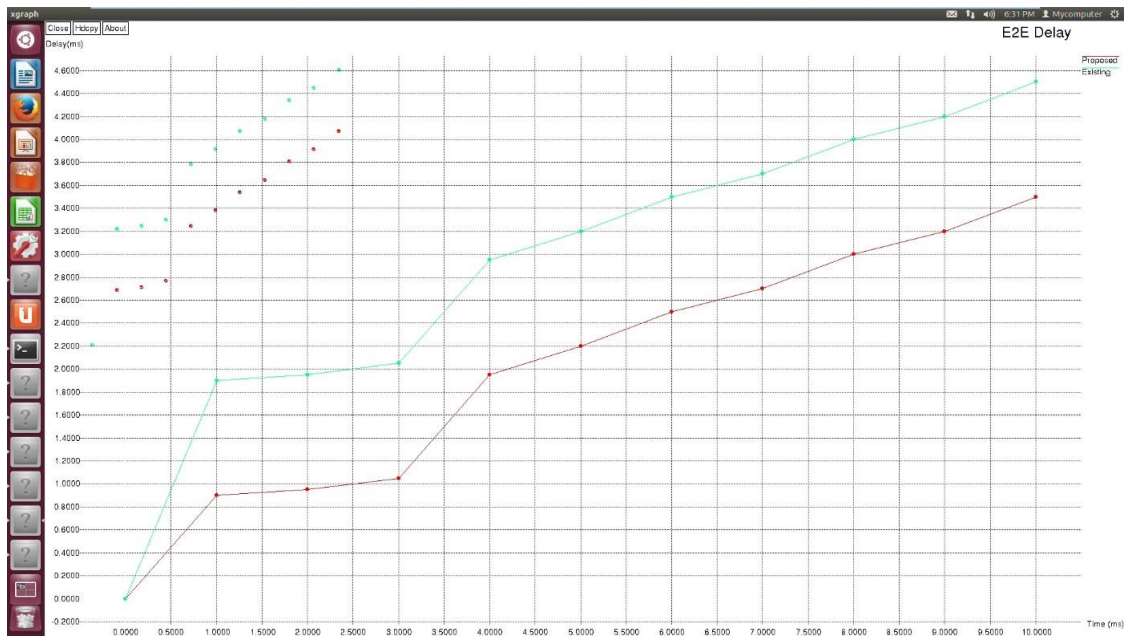


Fig. 5.13 End to End Delay

5.9.2 ECC Results

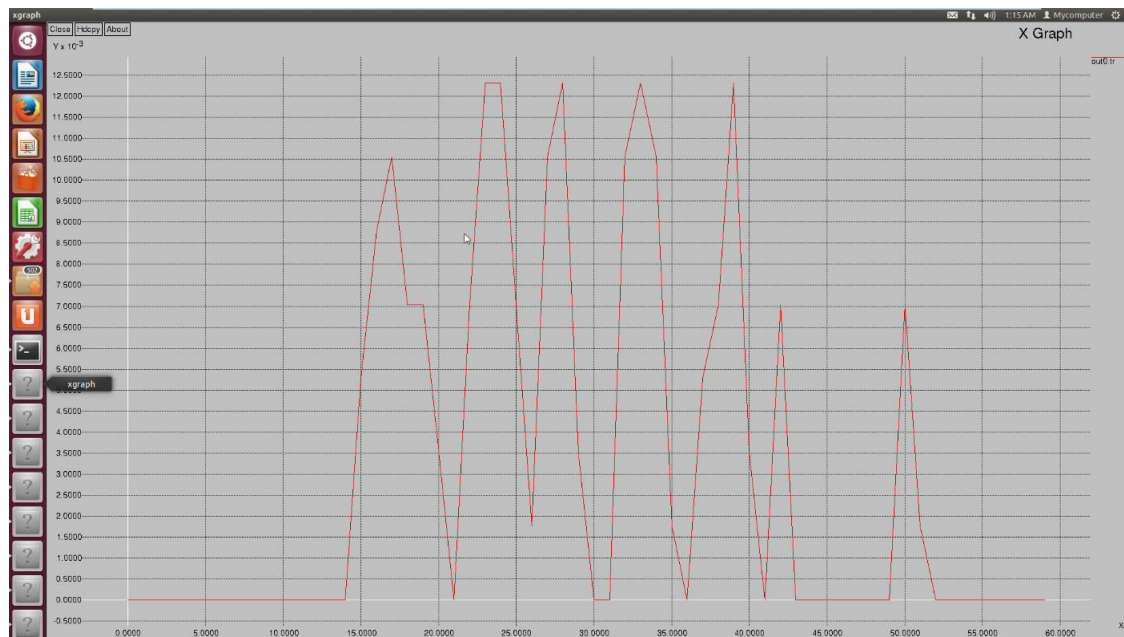


Fig. 5.14 Network Performance of ECC

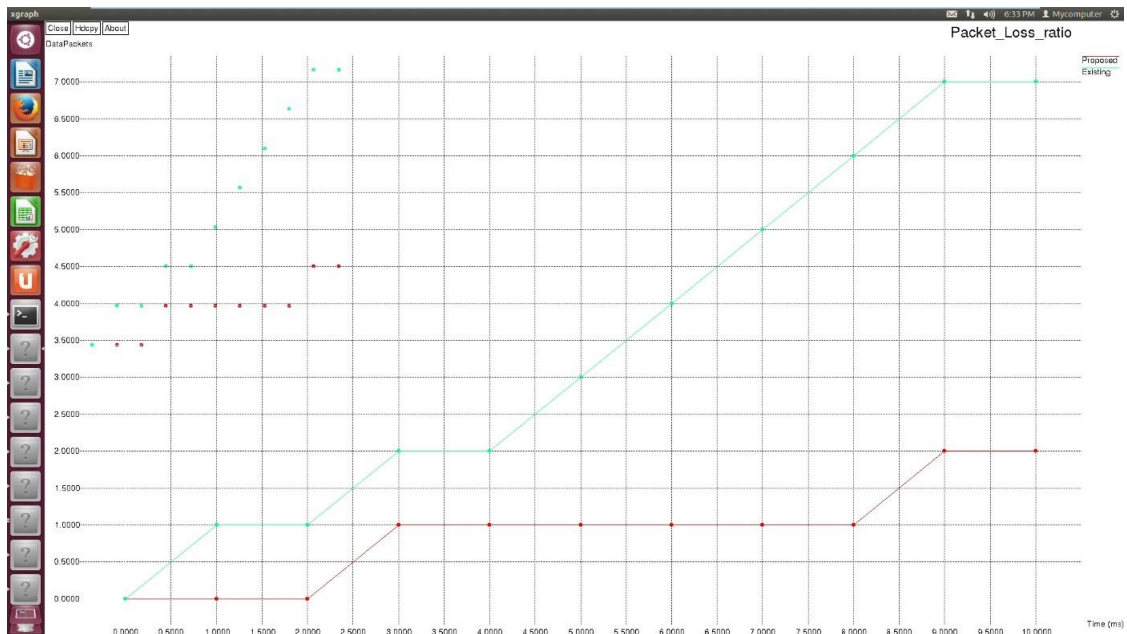


Fig. 5.15 Packet Loss Ratio

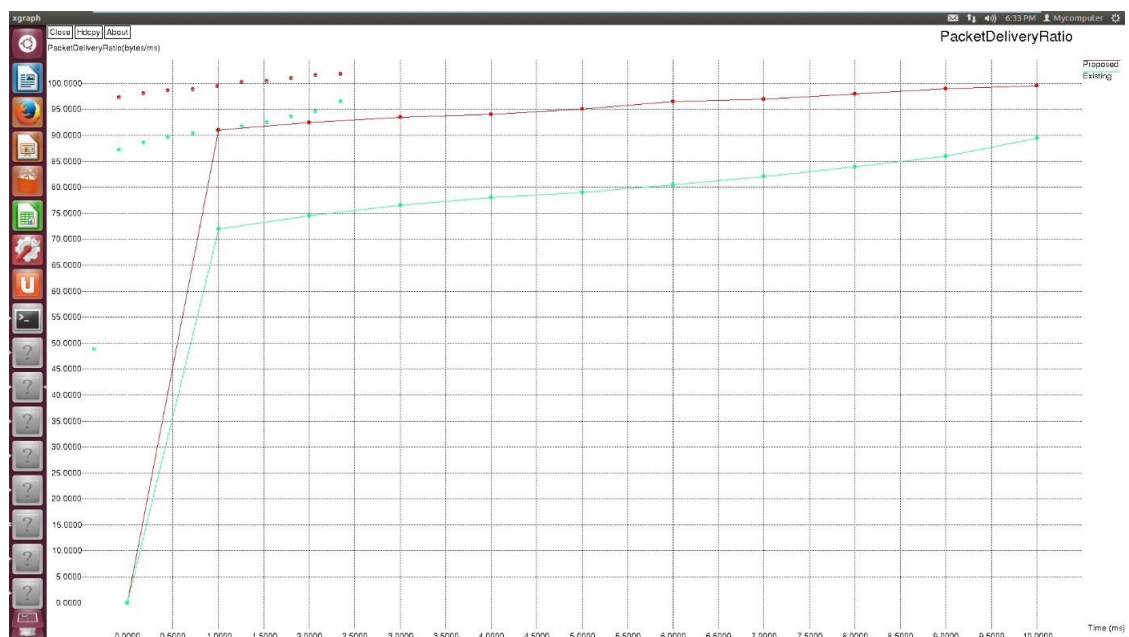


Fig. 5.16 Packet Delivery Ratio

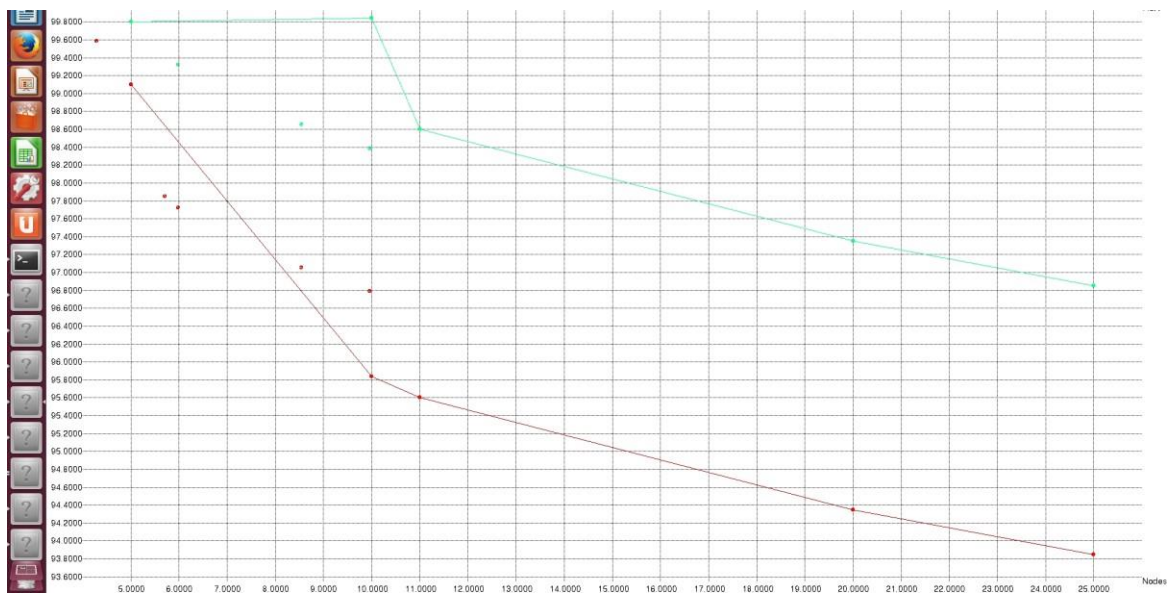


Fig. 5.17 Comparison of Energy

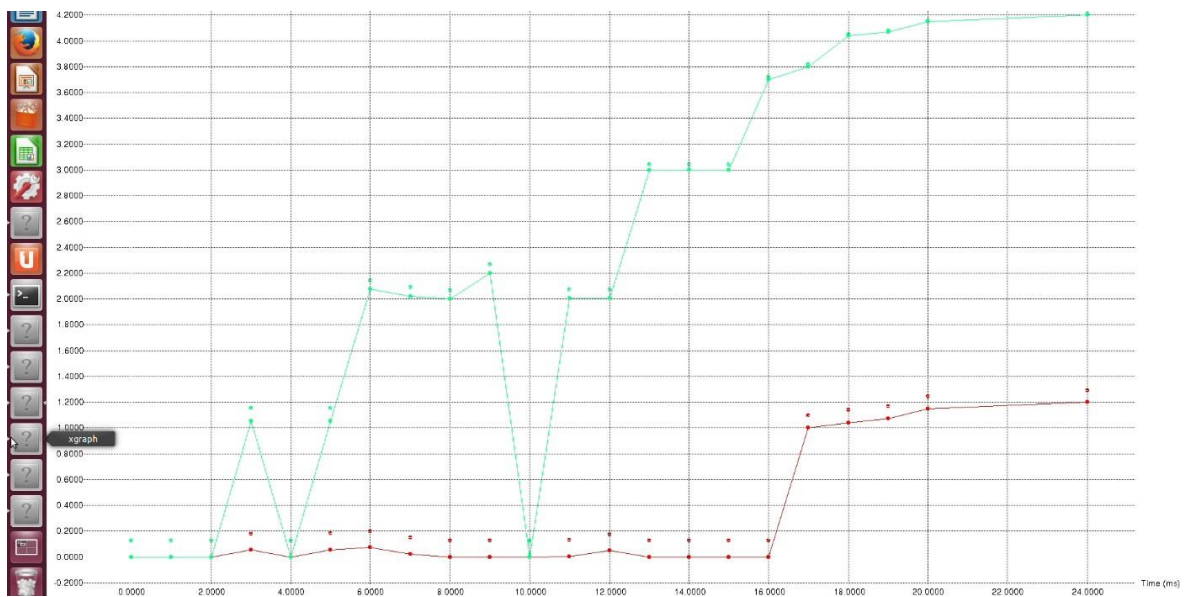


Fig. 5.18 Routing Overhead

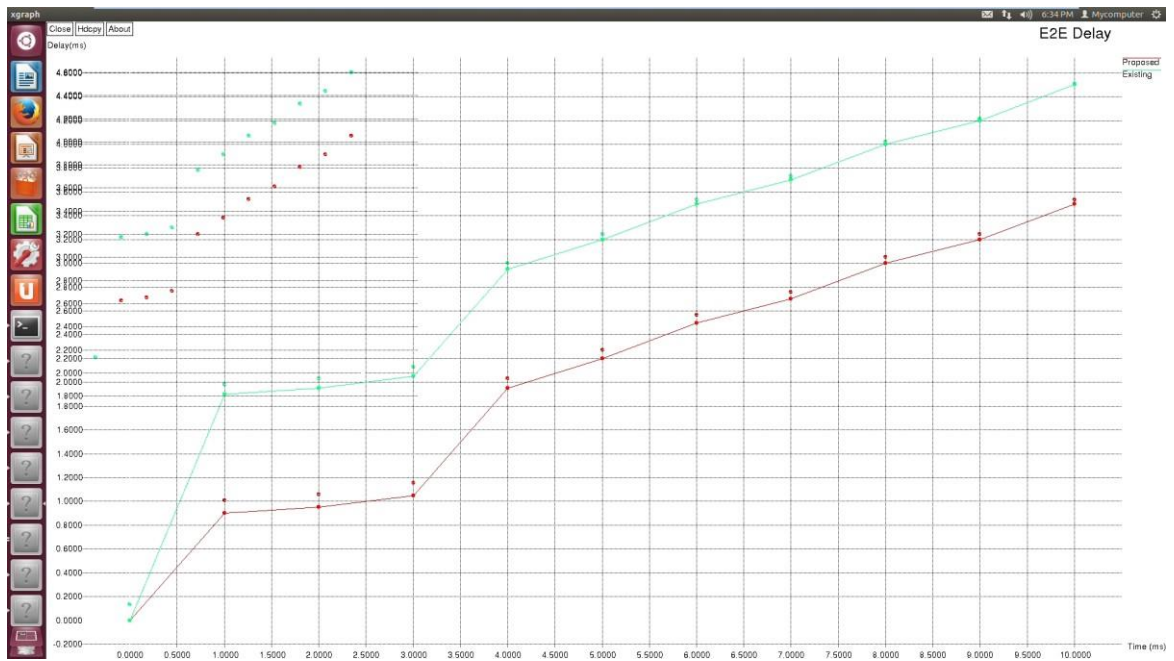


Fig. 5.19 End to End Delay

CHAPTER 6

SUMMARY AND CONCLUSION

Getting basic information move about the climate in military applications are essential since such information aid dynamic interaction. Late secure methodologies should be followed for moving the information. The got approaches utilized for moving the top-k question bring about the proposed work help with moving the basic information's safer in the multitier design of WSN. It in this way brings about accomplishing the outcome uprightness and fulfillment confirmation which is fundamental in numerous sensor related applications specifically modern non military personnel zones, territory checking and climate observing.

REFERENCES

- [1] Al-Fuqaha, A., Guizani, M., Mohaßmmadi, M., Aledhari, M. and Ayyash, M. "Internet of things: A survey on enabling technologies, protocols and applications," in IEEE Communications Surveys Tutorials, vol. 17, no. 4, 2015, pp. 2347-2376.
- [2] Brandt A. and Buron, J. "Transmission of IPv6 Packets over ITU-T G.9959 Networks," IETF RFC 7428, February 2015, <https://www.ietf.org/rfc/rfc7428.txt>, (accessed February 24, 2017).
- [3] Cetinkaya O. and Akan, O. "A dash7-based power metering system," in 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015, pp. 406-411. [15] HomePlug Alliance, "HomePlug™ AV2 Technology," 2007 http://www.homeplug.org/media/filer_public/2c/32/2c327fc8-25bb409e-abf7-c398534c24dc/homeplug_av2_whitepaper_130909.pdf, (accessed February 24, 2017).
- [4] Decuir, J. "Bluetooth 4.0: Low Energy," 2010, <https://californiaconsultants.org/wp-content/uploads/2014/05/CNSV-1205-Decuir.pdf>, (accessed February 24, 2017).
- [5] Gomez, C., Oller, J. and Paradells, J. "Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology," in Sensors, vol. 12, no. 9, 2012, pp. 11734-11753.
- [6] Granjal, J., Monteiro, E. and Sa Silva, J. "Security for the internet of things: A survey of existing protocols and open research issues," in IEEE Communications Surveys Tutorials, vol. 17, no. 3, 2015, pp. 1294-1312.
- [7] Hasan, M., Hossain, E. and Niyato, D. "Random access for machine- to-machine communication in lte-advanced networks: issues and approaches," in IEEE Communications Magazine, vol. 51, no. 6, 2013, pp. 86-93.
- [8] IEEE802.15.4-2011, "IEEE standard for local and metropolitan area network-part 15.4: Low-rate wireless personal area networks (LR-WPAN)," in IEEE Standards, April, 2012, pp.1-225.
- [9] ITU-T, "Short range narrow-band digital radio communication transceivers PHY and MAC layer specifications," 2012, <https://www.itu.int/rec/T-REC-G.9959>, (accessed February 24, 2017).

- [10] Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F. and Alonso- Zarate, J. "A survey on application layer protocols for the internet of things," in Transaction on IoT and Cloud Computing, vol. 3, no. 1, 2015, pp. 11-17.
- [11] Kim, A., Hekland, F., Petersen, S. and Doyle, P. "When hart goes wireless: Understanding and implementing the wirelesshart standard," in IEEE International Conference on Emerging Technologies and Factory Automation, 2008, pp. 899-907.
- [12] Nobre, M., Silva, I. and Guedes, L. A. "Routing and scheduling algorithms for WirelessHART Networks: a survey," in Sensors 15, no. 5, 2015, pp. 9703-9740.
- [13] Park, M. "IEEE 802.11ah: sub-1-ghz license-exempt operation for the internet of things," IEEE Communications Magazine, vol. 53, no. 9, 2015, pp. 145-151.
- [14] Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., McCann, J. and Leung, K. "A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities," in IEEE Wireless Communications, vol. 20, no. 6, 2013, pp. 91-98.
- [15] Stamford, "Gartner's 2014 hype cycle for emerging technologies maps the journey to digital business," August 2014, <http://www.gartner.com/newsroom/id/2819918>, (accessed February 24, 2017).
- [16] Zigbee, "Zigbee resource guide," 2016, http://www.nxtbook.com/nxtbooks/webcom/zigbee_rg2016/#/0, (accessed February 24, 2017).
- [17] Z-Wave, "Z-wave protocol overview," April 2006, https://wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243_2_Z_Wave_Protocol_Overview.pdf, (accessed February 24, 2017).

Randomized Generation Technique for Secured Data Transmission by Cryptographic Mechanism

ORIGINALITY REPORT

8%

SIMILARITY INDEX

4%

INTERNET SOURCES

3%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to University of Central Lancashire

Student Paper

1%

2

Jibendu Kumar Mantri, Rajalaxmi Mishra, P. Gahan. "A Novel Encryption Scheme Using Hybrid Cellular Automata", 2019 International Conference on Information Technology (ICIT), 2019

Publication

1%

3

authorzilla.com

Internet Source

1%

4

Gusev, Dmitry. "Research on the Algorithm of the Trustful Communication over the Metrized Small World Distributed Data Storage System", International Journal of Soft Computing and Software Engineering, 2012.

Publication

1%

5

Nettem Gopi, Nare Thoshan Kumar Reddy, T. Sasikala. "Chapter 68 Predicting Bike Rental Based on Environment and Seasons",

1%

6	www.enggjournals.com Internet Source	1 %
7	B. Suresh, A. C. Santha Sheela. "Chapter 41 Finding Smelly or Non-smelly Using Injected and Revision Method", Springer Science and Business Media LLC, 2021 Publication	1 %
8	Submitted to ABV-Indian Institute of Information Technology and Management Gwalior Student Paper	<1 %
9	Submitted to University of Bath Student Paper	<1 %
10	117.211.100.42:8180 Internet Source	<1 %
11	arxiv.org Internet Source	<1 %
12	sathyabamauniversity.ac.in Internet Source	<1 %
13	www.sathyabama.ac.in Internet Source	<1 %
14	Submitted to University of Essex Student Paper	<1 %

15	www.ijmetmr.com Internet Source	<1 %
16	abcdcn.blogspot.com Internet Source	<1 %
17	Submitted to Indian Institute of Science, Bangalore Student Paper	<1 %
18	mailman.isi.edu Internet Source	<1 %
19	"Communications and Networking", Springer Science and Business Media LLC, 2018 Publication	<1 %
20	"The Art of Wireless Sensor Networks", Springer Science and Business Media LLC, 2014 Publication	<1 %
21	mafiadoc.com Internet Source	<1 %
22	Submitted to Myongji University Graduate School Student Paper	<1 %
23	Submitted to Oklahoma State University Student Paper	<1 %
24	Submitted to Universiti Teknologi Malaysia Student Paper	<1 %