Journal of Science and Technology
ISSN: 2456-5660 Volume 7, Issue 04 (JUNE 2022)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2022.v7.i04.pp60-87*

Hybrid Diagonal Transposition Cipher Model for Securing Data in Software Defined Networks

P.Mohana Priya[1]
[1](*Department of Information Technology, SASTRA Deemed to be University,Thanjavur,India)*
Corresponding Author: mohanapriyatce@gmail.com

To Cite this Article

P.Mohana Priya[1]"**"Hybrid Diagonal Transposition Cipher Model for Securing Data in Software Defined Networks"***", Journal of Science and Technology, Vol. 07, Issue 04, June 2022.*

Article Info
Received: 25-04-2022 *Revised: 11-05-2022* *Accepted: 02-06-2022* *Published: 27-06-2022*

Abstract:*Data is an important asset for every organization and hence this article is proposed to secure data from common breaches in software defined network. In this article, hybrid cipher model is proposed to safeguard the communication of data transmitted among the layers in software defined networks. The logic of hybrid cipher model is incorporated in software defined controller which encrypts open flow request and response messages. Software Defined Network is adapted for implementing hybrid cipher model as the network provides customizable platform and act as a unmanned security featured software controller. The proposed Hybrid Diagonal Transposition algorithm is incorporated with software defined wireless sensing node for encrypting user's data. Hence the unmanned security featured wireless sensing node is situation-aware, it detects malicious traffic flows and encrypts user's data. Hybrid Diagonal Transposition algorithm prevents data breaches in Software Defined Networks. Results are interpreted for various network and sensor metrics such as routing hops, participating node temperature, battery voltage, humidity, lights, received packets per node, number of network hops, power consumption, radio duty cycle, temperature of sensors, beacon interval, network hops, routing metric and the same work will be extended in future for comparative results.*

Key Word*:Software defined networks ; Cryptography ; Information Security ; Wireless Sensor Networks ; Transposition Cipher Model.*

I. Introduction

Data security[1] is considered very essential, especially remote controlling of unmanned installations for instances, biomedical[2], building infrastructure[3], border, critical location[4], and sensitive data transmission[5]. The wireless sensor nodes deployed should differentiate the neighbor from the intruder, by any means, when an intruder happens to receive a data from an authenticated node, the data information should not be breached. The node is provisioned with reconfiguration strategies without even a minimum intervention from manual operation. Hence the wireless sensing node is situational aware[6].

In order to have network customization, the programmable and flexible Software Defined Network (SDN)[7,8]is chosen to integrate the concepts of cryptography for securing data transmitted across the planes. SDN is a three-layered network architecture in which the bottom data plane consists of switches[9,10]. The central plane consists of SDN controllers[11] both in the form of commercial and open-source. The top application plane[12]consists of various SDN applications and to the specific for securing data from massive attack traffic flows, appropriate applications are incorporated such as throughput, bandwidth etc.

In SDN, programming interfaces for north-bound and south-bound planes are associated in which Open Flow (OF)[13] protocol is a south bound interface in which communications of data and control planes are transmitted

in the form of Open Flow Protocol (OFPT) request and response messages. Data is propagated between data plane[14], control plane[15] and application planes within the programming interface. Data breach[16] occurs when there is a communication of messages between data, control and application plane.

Security vulnerabilities[17] of SDN architecture includes threat vectors in data-control plane and control-application plane. Data breaches occurs in data plane switches[18], controller[19] and applications[20] incorporated in application plane. Some of the threat vectors include vulnerable data, control and application plane.

The unmanned environment of Software Defined Wireless Networks (SD-WSN)[21], where insignificant or no possibility of handling the deployed sensor nodes to configure or for any other purposes; hence, nodes will be randomly deployed and each manages on its own, including recharging of its circuitry components, recognizing neighbor, acquiring of data, processing, securing and eventually transmitting to the identified neighbor in the network.

It is very relevant and necessary to improve the existing cryptography technique and ensure proper configuration within each sensor node and made available in the unmanned environment. Hence the SD-WSN should be situational aware to reconfigure the framed security policies in its flow tables based on the incoming network traffic flows.

The primary objective of this work is to provide an improved cryptography technique to be embodied within unmanned wireless sensor node and the control plane of software defined wireless sensor node is provisioned with situation aware mechanism that provides the reconfigurable policy-based alerts to the data plane.

The contribution of this article is referred as HyDiag, which rely on the combination of substitution-transposition cipher techniques where substitution method is applied for the corner values whereas transposition method for the rest of the values. The proposed HyDiag algorithm provides data confidentiality between data and control planes in software defined programming interface and integrated with existing protocol techniques. HyDiag proposes a grid box as a frame for plain and cipher texts. The said control plane of software defined wireless sensor node is provisioned with situation aware mechanism that provides the reconfigurable policy-based alerts to the data plane.

This research work is organized as section 2 discusses about the existing literature review, section 3 details about the proposed hybrid diagonal transpositional cryptographic algorithm, section 4 deals with experimental network setup, section 5 details about interpretations of results obtained and its corresponding discussions, section 6 concludes with future research directions.

II. Literature Review

Almohaimeed, A and Asaduzzaman, A (2019) proposed Moving Target Defense (MTD)[22]to protect communication links from sensitive data leakages. MTD is intended to protect user identities contained in transmitted messages that prevents network intruders from identifying the real identities of senders and receivers. Mohana Priya proposed secure defense mechanism[23]which consists of Restricted Boltzmann Machine (RBM)[24]algorithm to detect Distributed Denial of Service (DDoS) attacks in SDN controller. Diffie-Hellman Key Exchange algorithm is implemented in OF protocol version 1.3 where random private keys are generated for every incoming network traffic flows.

III. Proposed Hybrid Diagonal Transposition Cryptographic Algorithm

The conceptual idea of the proposed HyDiag algorithm relies on the combination of substitution-transposition cipher techniques to provide data confidentiality between client-server and also for data storage platforms. The unique parameter incorporated in the proposed cipher model is that based on the key value, the plain text message is arranged in the grid boxes which enhance security with respect to data confidentiality comparatively than the existing cipher models. The sender begins encrypting by arranging the plain text message as per the positional value of the key. As a follow up, two encryption logics are applied for the grid boxes, to start with deploying caesar's method for the corner values and single shifting of plain text letters from bottom-top in the grid boxes. The cipher text can be read from the corners of the grid boxes diagonally and with row wise values. The cipher text is the merged result of first part of cipher text obtained from the first grid box and second part of cipher text is obtained from the second grid box. The receiver decrypts the cipher text by making use of reversing procedures of encryption. The proposed cipher model can be integrated with a wireless sensing node. The uniqueness of the proposed cipher model is about framing grid boxes for plain text, cipher text. Substitution method

is applied for the corner values whereas transposition method is applied for the rest of the values. Reading of values in a diagonal and row-wise bottom-up and top-down approach is novel ways of providing confidentiality as it is hard to crypt analyze which results in time consumption.

From the literature survey, it is observed that most of the existing transposition cipher models are relying of some patterns for arrangement of plain text and for reading encrypted and decrypted messages with the same pattern. The proposed HyDiag transposition cipher model makes use of caesar cipher model of encryption only for the corner values of the grid box which are segregated based on the key size and applied on the plain text to make the grids. In the legacy transposition techniques, same logic is applied for all constructed grid boxes which results with easiest brute forcing of key value used in encryption. Applying same logic for transpositioning plain text message itself becomes an attractive target for the attackers which is considered as a security loophole of the proposed HyDiag algorithm. Usage of two distinct cryptographic algorithms does not result with complexity either in terms of processing and time. The working procedure of HyDiag Transposition cipher model is given as an algorithm which is given as follows.

3.1 Pseudocode of Proposed Hybrid Diagonal Transposition Cipher Model

Step - 1: Write the plain text message in row wise of variable length size.

Step - 2: Generate ASCII key value and read the plain text message column by column based on the key positions.

Step - 3: Create grid boxes by dividing the count of key value by 2. If the resultant value is exactly the whole number, consider the value as number of columns in the first grid box and remaining plain text letters to be arranged in the second grid box.

Step - 4: In both the grids, for the corner values of grid box the logic of caesar cipher is applied both for encryption and decryption.

Step - 5: For the remaining plain text letters in the grid, shifting is done from bottom to top row by row until (last row - 1).

3.2 Proposed HyDiag Transposition Cipher Model

3.2.1Algorithm of Proposed HyDiag Transposition Cipher Model

Plain text Message: CANCELOPERATIONDBYCOMMANDO Key: 5312476

Step - 1: Arrangement of plain text message in row wise.

Step - 2: Transpositioning plain text letters based on the key value (5312476)

Step - 3: Divide the plain text message into two halves by dividing the key with value 2, if the resultant value is a whole number, it can be considered as such for creating the first grid box, the resultant value is decided as the number of columns and the remaining plain text letters can be arranged in the second grid box and so on. For the plaintext **CANCEL OPERATION D BY COMMANDO,** step – 3 results with the following grid boxes.

C	A	N	C	E	L	O
P	E	R	A	T	I	O
N	D	B	Y	C	O	M
M	A	N	D	O	X	X

GridBox 1 (GB1)

E	N	C	A	C	O	L
T	R	P	E	A	O	I
C	B	N	D	Y	M	O
O	N	M	A	D	X	X

GridBox 2 (GB2)

Algorithm 1 discusses about the encryption procedure of HyDiag cipher model, where it is applied to the message communications between data and control planes. Hence all the Open Flow (OF) request and response messages will be encrypted which prevents data leakage attacks in SDN. The encryption algorithm is fed with plain text, key and number of columns and it is used at the sender's end.

Algorithm 1: HyDiag Encryption

Input: Plain Text, Key, Number of Columns as per key value

Output: Cipher Text

1 Initialize Plain text Message (P.T)

2 Initialize Key Value (K)

3 Initialize Corner Values (CV)

4 Initialize Diagonal Values (DV)

5 Initialize Row Values (RV)

6 **Construction of Grid Boxes (GB1) and (GB2)**

7 No. of. Columns in GB1 = R(K/2)

8 **if K is 7**, an Odd Number

9 No. of. Columns in GB1 = R(7/2)

10 No. of. Columns in GB1 = 4 and No. of. Columns in GB2 = 3

11 else

12 No. of. Columns in GB1 and No. of. Columns in GB2 is equal

13 **Arrangement of Plain text letters in Grid Boxes**

14 **Apply Caesar Cipher logic in the Corner Values (CV) of grid boxes**

15 E(CV(GB1),CV(GB2)) = CV(GB1,GB2) + 3

16 **if CV in GB**

17 Apply Caesar Logic

18 else

19 One shift from bottom to top

20 **C.T = Read DV(GB1) + RV(GB2)**

Algorithm 2 discusses about the decryption procedure of HyDiag cipher model where all the encrypted Open Flow (OF) request-response messages are decrypted and it is used at the receiver's end. HyDiag decryption algorithm is fed with cipher text, key and number of columns.

Algorithm 2: HyDiag Decryption

Input: Cipher Text, Key, Number of Columns as per key value

Output: Plain Text

1 Initialize Cipher text Message (C.T)

2 Initialize Key Value (K)

3 **Construction of Grid Boxes (GB1) and (GB2)**

4 No. of. Columns in GB1 = R(K/2)

5 **if K is 7, an Odd Number**

6 No. of. Columns in GB1 = R(7/2)

7 No. of. Columns in GB1 = 4 and No. of. Columns in GB2 = 3

8 else

9 No. of. Columns in GB1 and No. of. Columns in GB2 is equal

10 **Arrangement of Cipher text letters in Grid Boxes**

11 **Apply Caesar Cipher logic in the Corner Values (CV) of grid boxes**

12 D(CV(GB1), CV(GB2)) = CV(GB1,GB2) - 3

13 **if CV in GB**

14 Apply Caesar Logic

15 **else**

16 One shift from top to bottom

17 **Plain Text (P.T) = Read DV(GB's) + RV(GB's)**

18 **Match with CV in GB**

19 **Read GB's in row wise to obtain plaintext**

Step - 3.1: Here the key count value is 7, when divided by 2, results 3.5, as it is a floating-point value, 3.5 can be considered as 4. The value 4 is none other than number of columns in the first grid box. Remaining trans positioned plain text letters can be arranged in the second grid box. The constructed grid boxes are shown below.

E	N	C	A

T	R	P	E
C	B	N	D
O	N	M	A

C	O	L
A	O	I
Y	M	O
D	X	X

Step - 4: Encryption Process

Step - 4.1: Apply the logic of caesar cipher model both for encryption (P.T + 3) and decryption only for the corner values of the grid box.

E+3 = H	N	C	A+3=D
T	R	P	E
C	B	N	D
O+3=R	N	M	A+3=D

C+3=F	O	L+3=O
A	O	I
Y	M	O
D+3=G	X	X+3=A

Step – 4.2: Other than the plain text letters in the corner of the grid, the remaining grid values can be single shifted from bottom to top for enhancing security in the traditional diagonal transposition cipher model.

H	N	M	D

N	R	P	C
T	B	N	E
R	C	D	D

F	X	O
O	O	A
Y	I	O
G	M	A

Step – 5: Cipher text can be framed by reading of the diagonal values from the two corners of the grid from left to right.

H			D
	R	P	
	B	N	
R			D

F		O
	O	
Y		O

	N	M	
N			C

T			E
	C	D	

	X	
O		A
	I	
G	M	A

Step - 6: First part of the cipher text is framed by reading the diagonal values of the grid box and second part of the cipher text is framed by reading the remaining shifted plain text letters from the grid box. Merging both grid box values can be the complete cipher text (C.T).

Step - 6.1: Diagonal values and shifted plain text letters of first grid box is represented using the term C1.

Step - 6.2: Diagonal values and shifted plain text letters of second grid box is represented using the term C2.

$$C.T = C1 + C2 \quad (1)$$

First part of Cipher text from the first grid box

(C.T1) = HRNDDPBRNMNCTECD

Second part of Cipher text from the second grid box (C.T2) = FOOOOYXOAIGMA

C.T = C.T1 + C.T2 = HRNDDPBRNMNCTECDFOOOOYXOAIGMA

Step - 7: Decryption Process

Step - 7.1: Arranging cipher text letters in the grid diagonally and for decryption process the corner values of the grid can be applied with the logic of caesar cipher, (C.T - 3) followed by single shifting of remaining plain text letters in top-down approach results the following grid boxes.

H-3=E	N	M	D-3 =A
N	R	P	C
T	B	N	E
R-3=O	C	D	D-3=A

F-3=C	X	O-3=L
O	O	A

Y	I	O
G-3=D	M	A-3=X

E	N	C	A
T	R	P	E
C	B	N	D
O	N	M	A

C	O	L
A	O	I
Y	M	O
D	X	X

Match the grid box values with the actual plain text values and arrange according to the key values which results the following grid as,

C	A	N	C	E	L	O
P	E	R	A	T	I	O
N	D	B	Y	C	O	M
M	A	N	D	O	X	X

Step - 8: Read the values in the grid box in row wise to obtain the plain text message. Here, the obtained plain text message is CANCELOPERATIONDBYCOMMANDO and the grid ended with two filler letters at the end of grid box.

IV. Experimental Setup

The wireless sensor network topology is designed using cooja simulator [25], and encryption, decryption logic is programmed as Contiki with (.csc) extension. The designed network topology runs on Sky mote firmware,

where Graphical User Interface (GUI) of the Collect View module of the mote is enabled in which command for each and every node is executed. The contiki encryption and decryption programs is considered as a process in thread and the system characteristics are defined using sensors and network metrics.

The specifications of all hardware components are given below. For implementing HyDiag transpositional cipher model, sky mote is selected to run the firmware process and observed sensor and network data during transmission and reception of data during encryption and decryption process. Once the contiki operating system is booted, radio medium (or) radio model is selected as Unit Disk Graphic Medium (UDGM), Distance Loss with transmission delay metric. As a follow up, sky mote is selected in which two firmware processes are uploaded to execute the HyDiag transposition cipher model and to visualize nodes incorporated in motes using collect view module. Finally commands to the nodes have been sent to observe sensor and network data where sensor data includes temperature, average temperature, battery with respect to voltage, power consumption of nodes, average power consumption, LED lights and humidity and relative humidity. Network data includes next hop information, average routing metric, beacon intervals, latency, Low Power Mode (LPM) and Central Processing Unit (CPU). Serial Flash Memory consists of processes executed and embedded antenna is attached for transmitting and receiving the sensor data. The specifications of Tmote Sky is given with the details of Telos. B/Tmote Sky such as Micro Controller Unit (MCU), TI MSP 430, Up to 8 MHz clock, 16-bit Flash memory (program): 48 k.B, SRAM (data and variables): 8 k.B Serial flash memory: 1 MB Sensors (installed on Telos. B only), Light, IR, Humidity, Temperature Radio (embedded antenna), CC 2420: 2. 4 GHz (IEEE 802. 15. 4), Max. data rate: 250 kbits/s.

Sensor metrics include average temperature, temperature, battery voltage, battery indicator, relative humidity and LED lights, power metrics include average power, radio duty cycle, instantaneous power and power history whereas network metrics include ETX (overtime), next hop (over time), latency, Received (Over Time), Lost (Over Time), Received (per Node), Received (5 mins), Neighbors, Beacon Interval, Network Hops (Over Time), Network Hops (Per Node), Routing Metric (Over Time), Average Routing Metric (Over Time). Each node includes Packets received per Node, Packets Lost, Number of Hops, Routing Metric, Beacon Interval, Reboots, CPU Power, LPM Power, Listen Power, Transmit Power, Power, On-time, Listen Duty Cycle, Transmit Duty Cycle. Node details include Packet Interval, Packet Randomness, Hop-by-Hop retransmissions, Number of reports. The network topology runs with encryption and decryption algorithm and during program execution the sensor and network related parameters are analyzed.

V. Results and Discussions

Figure 5.1 shows GUI of the Cooja simulator, in which the HyDiag encryption and decryption code is uploaded. Sky Mote is selected as a firmware to upload the scripts and start the simulation. All network and sensor metrics considered in this section are the default metrics provided by the cooja simulator.

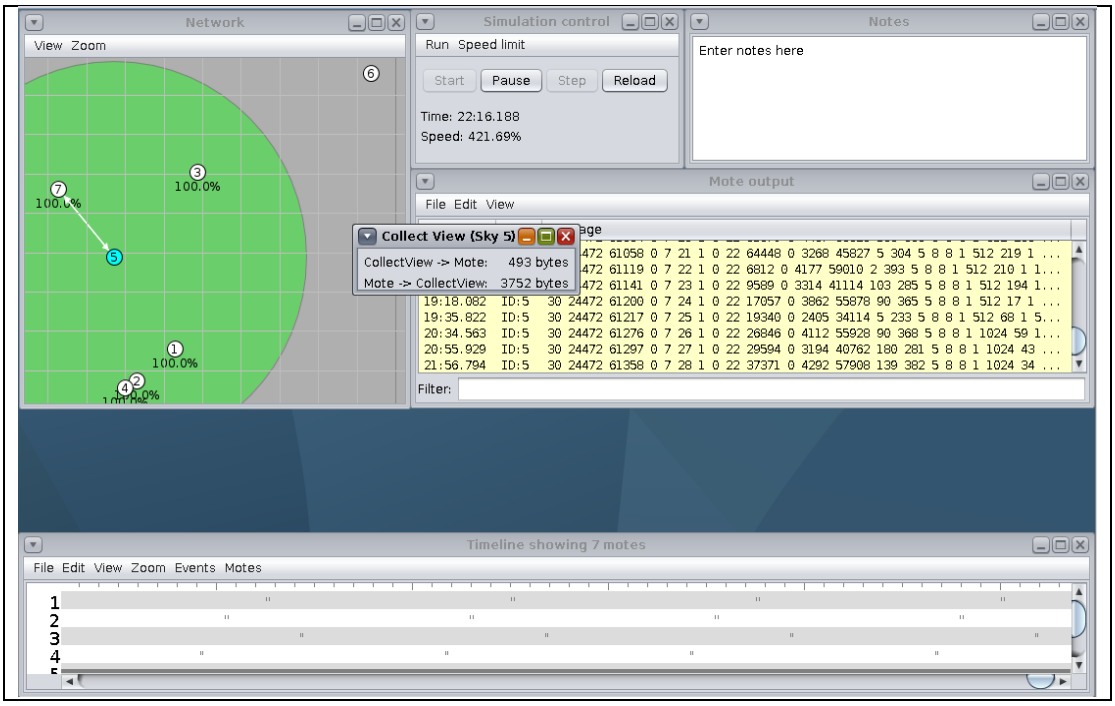


Figure 5.1: Uploading HyDiag Encryption and Decryption Code in Cooja Simulator

Figure 5.2 shows the node control interface in which randomness about the node to receive packets, hop-by-hop retransmissions, packet report interval is defined. The node control interface is visualized by selecting the specific node and to access collect viewfile.

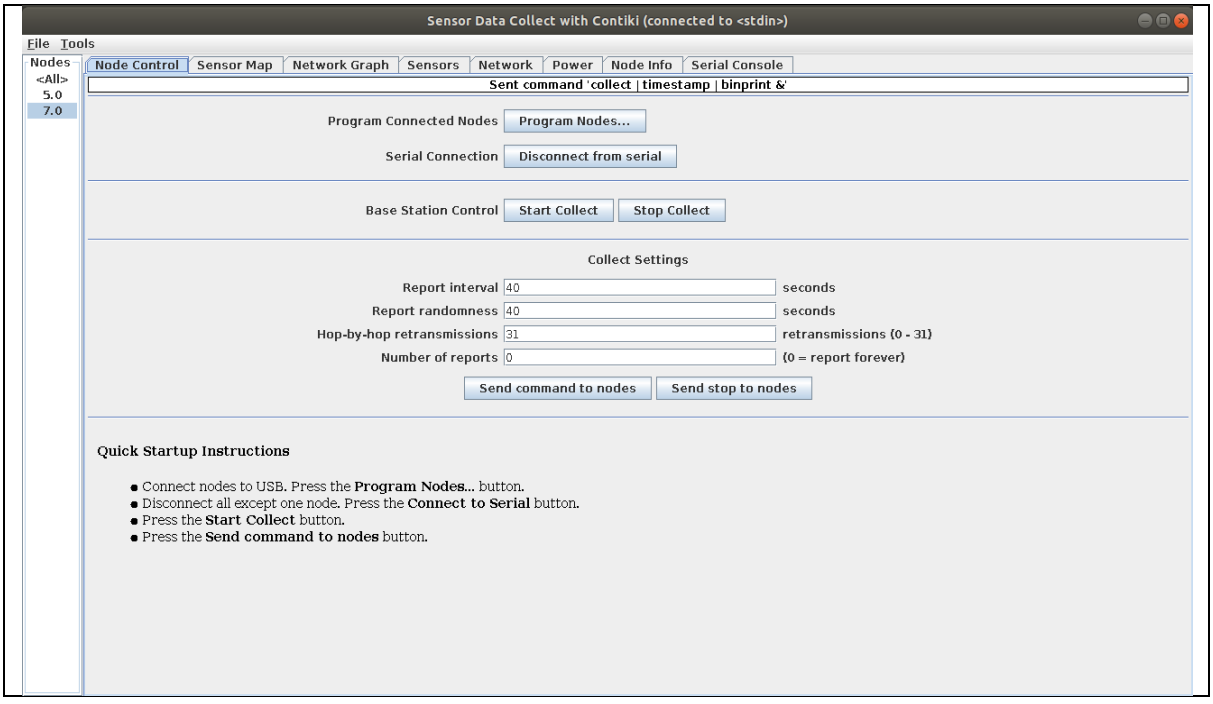


Figure 5.2: Sensor Data Collection on the selected Node

Figure 5.3.1 shows the sensor related inferences such as temperature, average temperature, battery voltage, battery indicator, relative humidity, light 1 and light 2. Temperature is measured for the number of nodes available with the measurement Celsius. It is observed that for the HyDiag encryption code, constant temperature is observed without deviations and the static temperature value is found to be 619.0 Celsius as in figure 5.3.1 and average temperature is shown in figure 5.3.2. Some deviations are found for the battery voltage from the 0.0 to 0.4 as in figure 5.3.3, the highest deviation value found is 0.4 from the inference. Battery indicator value is found to be 1 as in figure 5.3.4, humidity as 100 in figure 5.3.5, light 1 deviation value ranges from 277 to 175 as the first range, 175 to 345 as the second range, 345 to 195 as the third range, 195 to 300 as the fourth range, 300 to 0 as the fifth range, 0 to 370 as the sixth range, 370 to 315 as the seventh range, 315 to 60 as the eighth range, 60 to 0 as the ninth range and 0 to 350 as the tenth range as in figure 5.3.6 and frequent deviations found in figure 5.3.7.

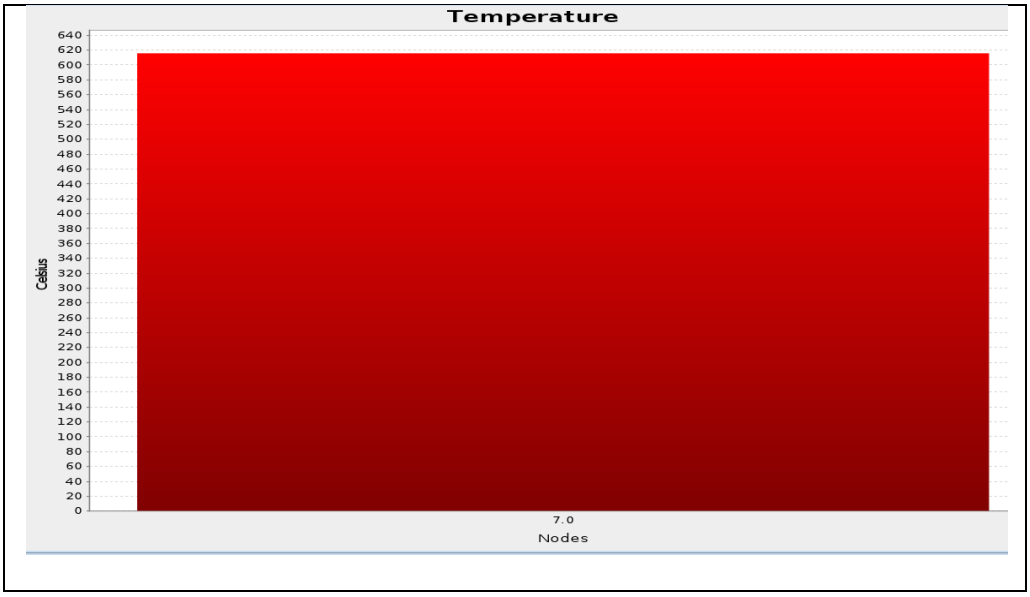


Figure 5.3.1 Participating Node Temperature

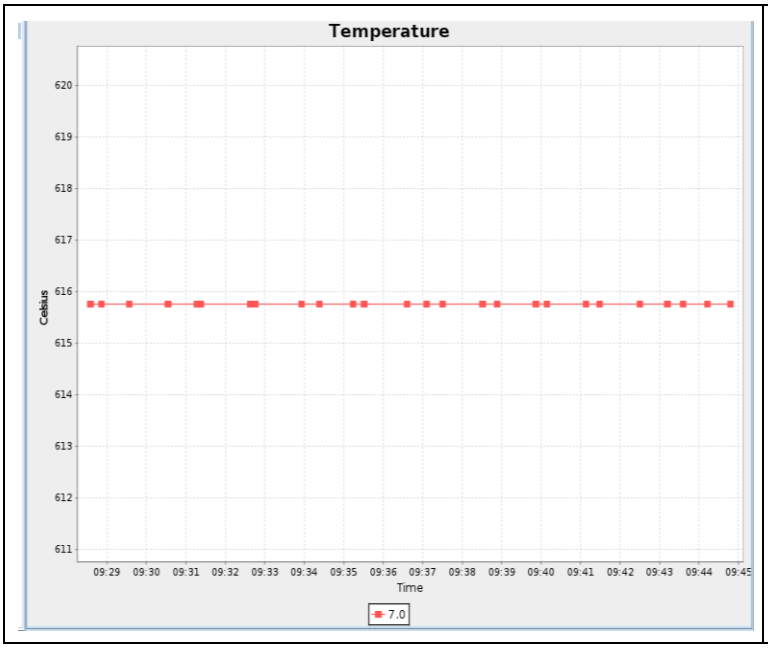


Figure 5.3.1: Average Temperature of Sensors

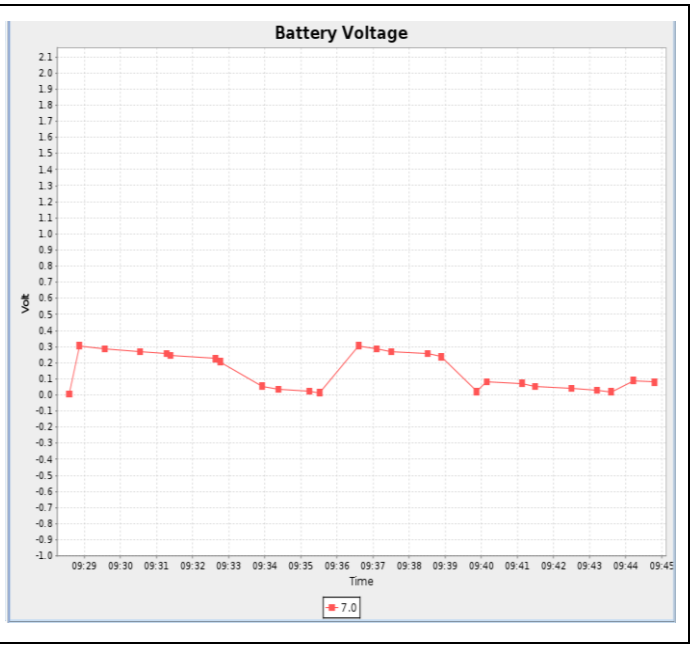


Figure 5.3.2: Battery Voltage

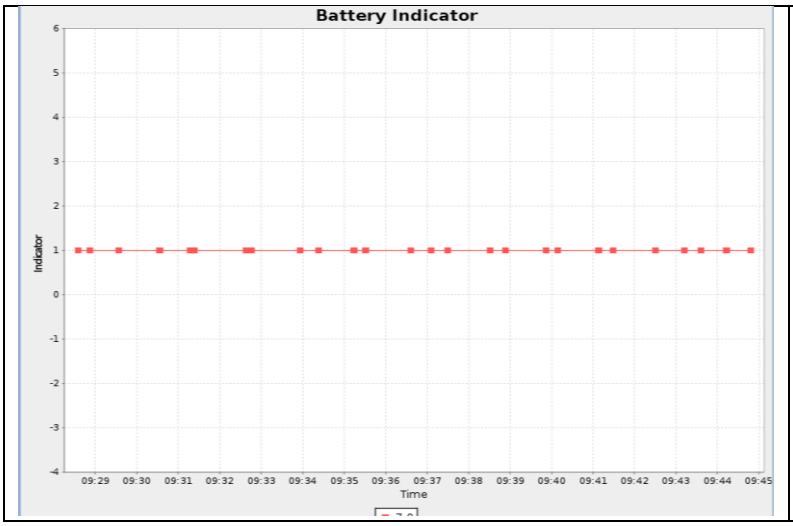


Figure 5.3.3: Battery Indicator

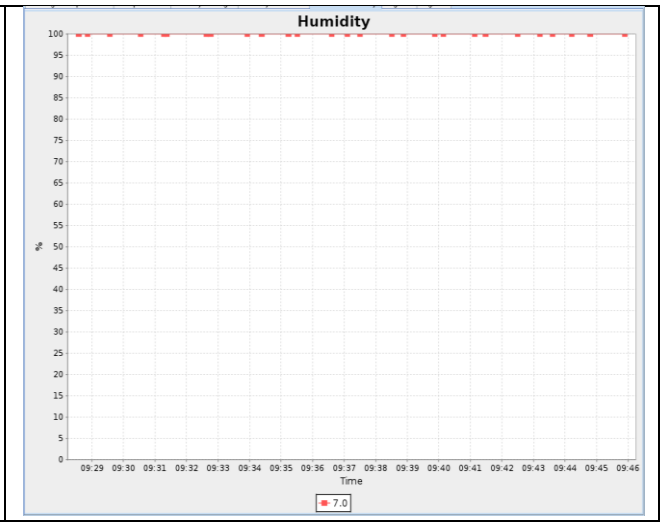


Figure 5.3.4 : Humidity

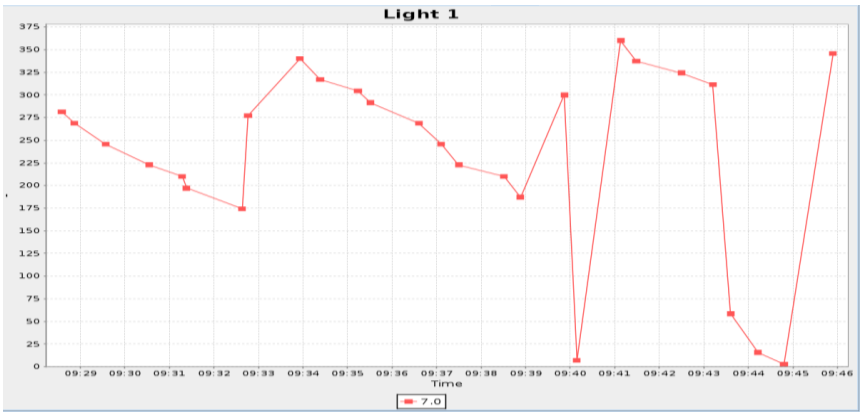


Figure 5.3.5: Inference of Light 1

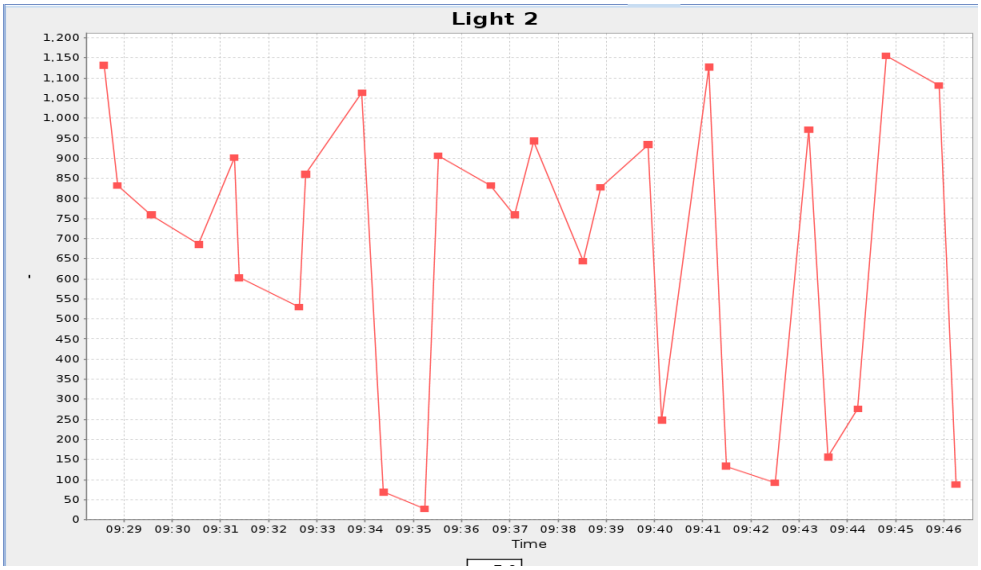


Figure 5.3.6: Inference of Light 2

Figure 5.4.1shows the network metrics such as received packets per node, received packets on all the participating hosts as in figure 5.4.2, network hops per node as in figure 5.4.3 which consist of average hops and last hops information.

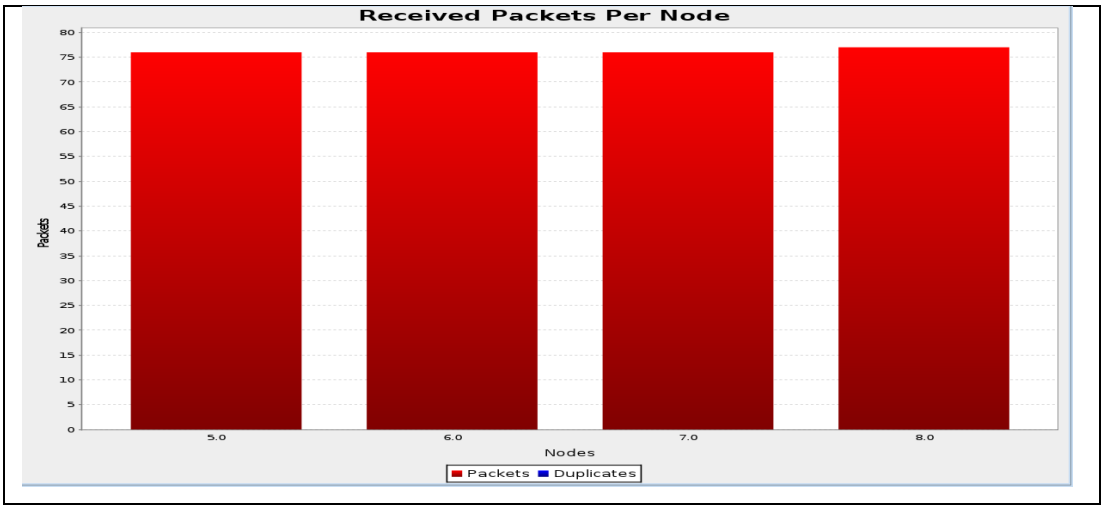


Figure 5.4.1 Received Packets Per Node

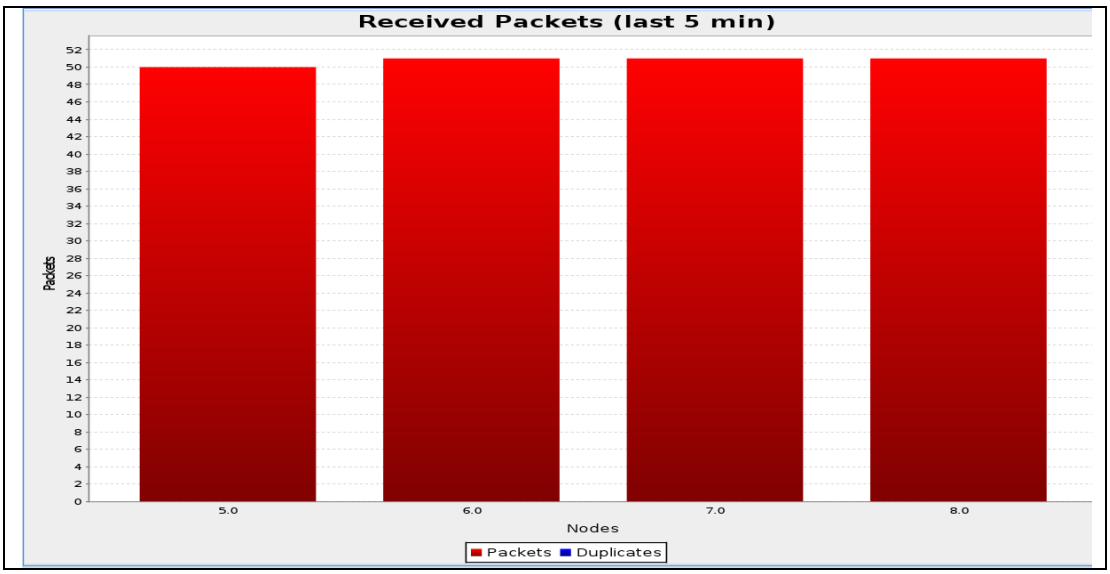


Figure 5.4.2 Received Packets on the Participating Nodes

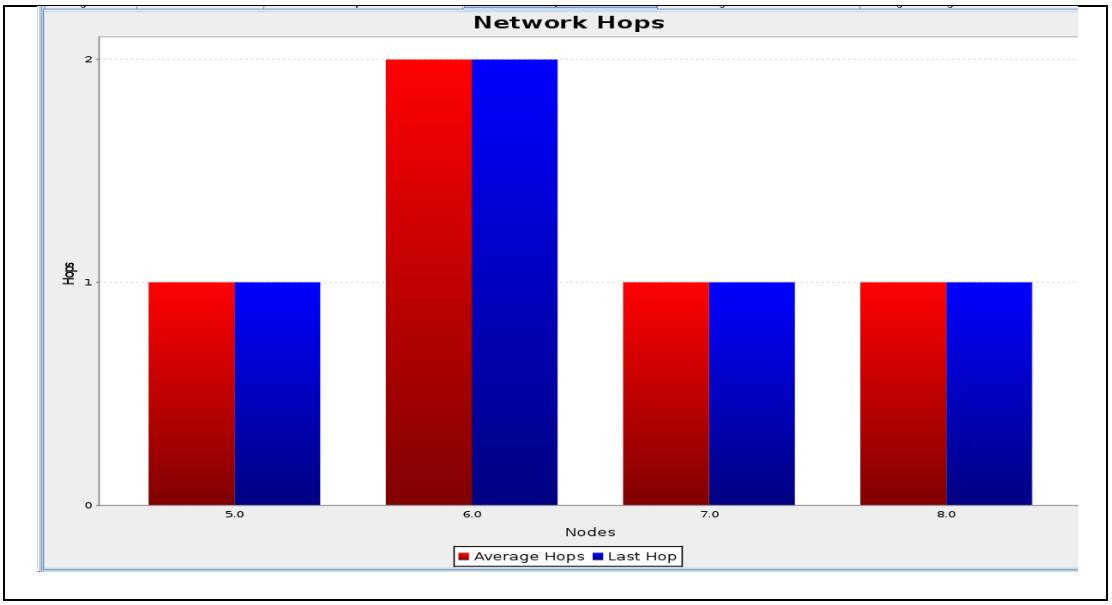


Figure 5.4.3 Number of Network Hops

Figure 5.5.1 shows the power consumption of sensors deployed in sky mote with metrics such as LPM, CPU, Radio listen and Radio transmit. These metrics are measured as megawatts (mW). LPM values for the simulated sensor node ranges between 0 to 0.15, for CPU it is 0.15 to 0.50, Radio listen from 0.50 to 0.89 and radio transmit values ranges from 0.89 to 0.92. Figure 5.5.2 shows the radio listen and radio transmit values for the wireless sensor nodes ranges between 0 to 0.625 and from 0.625 to 0.710 and it is measured as duty cycle in percentage and Figure 5.5.3 illustrates the instantaneous power consumption of LPM, CPU, Radio listen and Radio messages transmit. It is observed in the inference that among the four transmitting sensor nodes, node 8 consumes high power in case of data transmission which is in the range of 0.87 to 1.00.

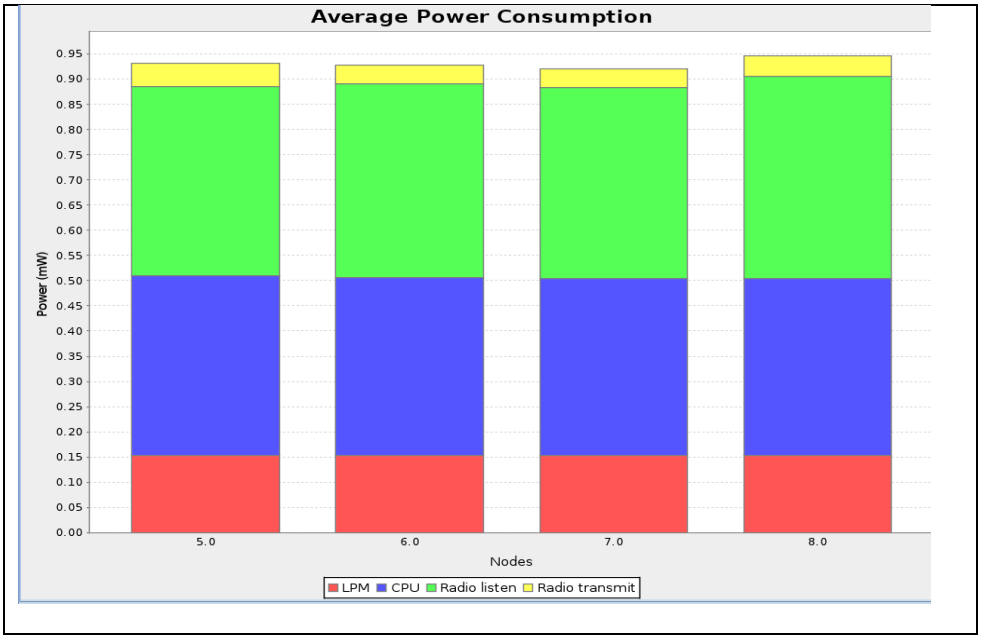


Figure 5.5.1 Average Power Consumption

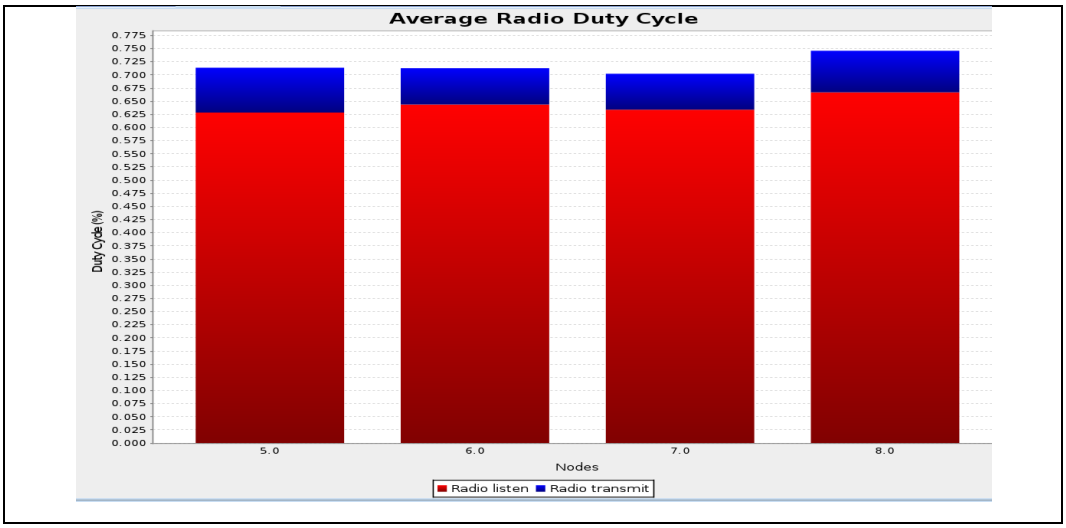


Figure 5.5.2 Average Radio Duty Cycle for Radio Listen and Radio Transmission

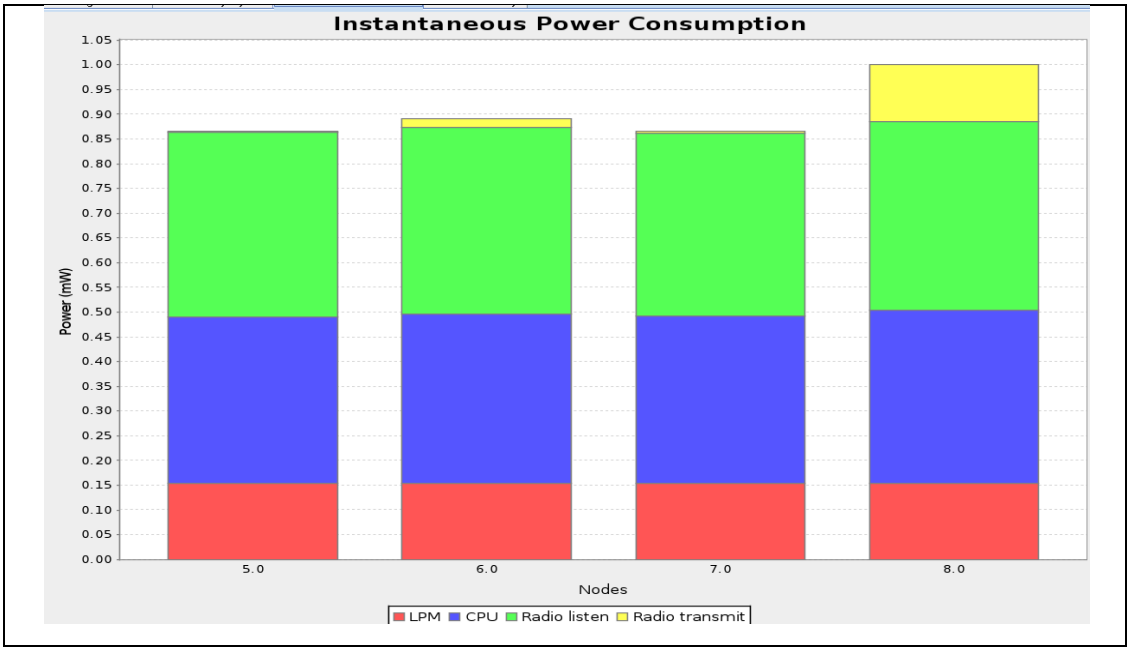


Figure 5.5.3 Instantaneous Power Consumption

Table 5.1.1and 5.1.2 shows the sensor node information deployed on the mote such as participating nodes, received packets, number of hops, routing metric, beacon interval, CPU power, LPM power, Listen Power, Transmit Power, Power consumption, Listen Duty Cycle and Transmit Duty Cycle.

Table 5.1.1: Details of Sensor Node

Node	Received	Hops	Rtmetric	Beacon	CPU Power

	Packets			**Interval**	
5.0	78	1.000	10.179	27 mins, 28 sec	0.356
6.0	78	2.000	17.538	27 mins, 08 sec	0.353
7.0	78	1.000	9.538	27 mins, 10 sec	0.351
8.0	78	1.000	8.000	27 mins, 17 sec	0.352
9.0	0	0.000	0.000	0.000	0.000
Avg	**78.000**	**1.250**	**11.314**	**27 mins, 16 sec**	**0.353**

Table 5.1.2: Details of other parameters of Sensor Node

LPM Power	**Listen Power**	**Transmit Power**	**Power**	**Listen Duty Cycle**	**Transmit Duty Cycle**
0.153	0.377	0.045	0.931	0.628	0.084
0.153	0.386	0.036	0.928	0.644	0.068
0.153	0.380	0.036	0.921	0.634	0.068
0.153	0.400	0.043	0.948	0.667	0.081
0.000	0.000	0.000	0.000	0.000	0.000
0.153	0.386	0.040	0.932	0.643	0.075

Figure 5.6.1 shows the sensor related inferences such as temperature, average temperature, battery voltage, battery indicator, relative humidity, light 1 and light 2.

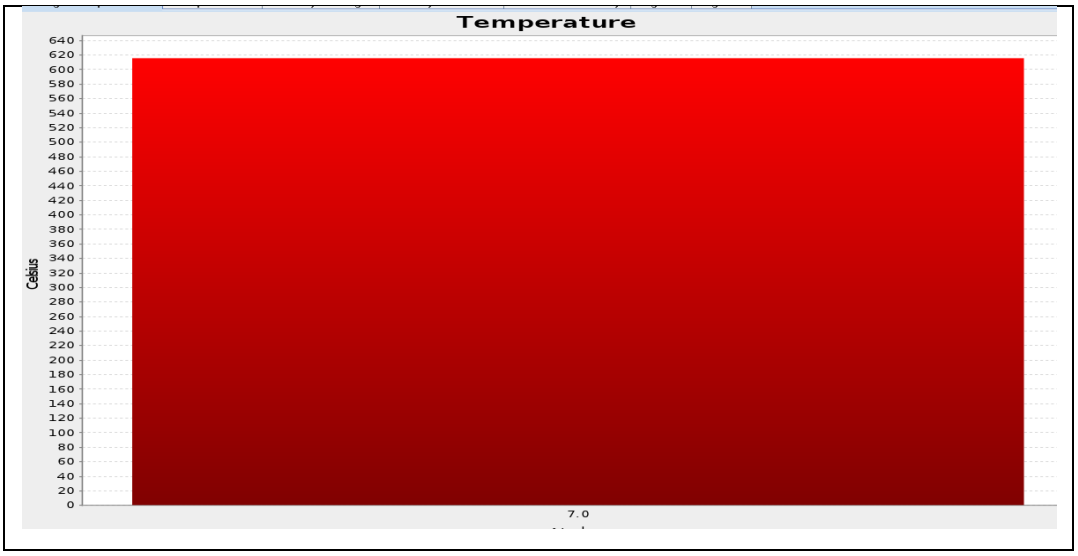


Figure 5.6.1: Temperature of Sensors

Figure 5.7.1 shows the number of packets received per node. It is found that while encryption module is executed, 29 packets are transmitted by other nodes such as 5, 6 and 8 to node 7.

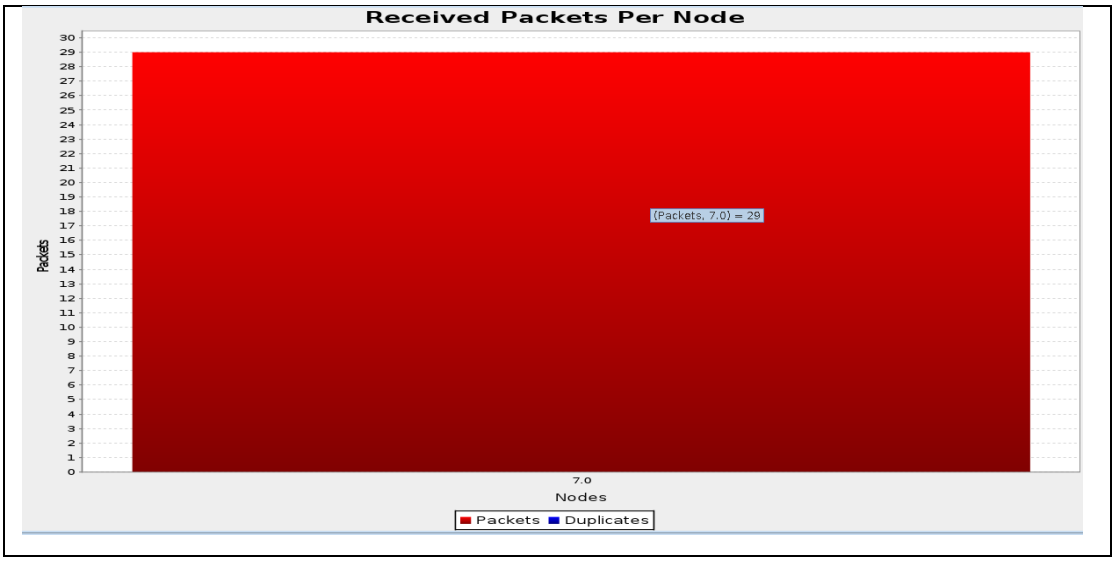


Figure 5.7.1 Received Packets Per Node

Figure 5.8.1shows the illustration of received packets per node for node 7, in which the total duration of the execution of encryption module is 10 minutes where number of packets is constantly increased. Figure 5.8.2shows the transmission of messages which is measured in terms of beacon intervals, for **9.29th**, **9.30th**and **9.31st**seconds, 250 beacon intervals for the transmission of packets is observed for the node 5, node 6, node 7, node 8 and from the time frame of **9.33rd** second to **9.41st** second, beacon interval is found as 550 for the transmission of packets. For the time frame from **9.41st** to **9.48th** second, beacon interval is found as 1025 for the packet transmissions. The next hop node is found as node 1 as it is a monitoring node and average hop and in the collect view module of the sky mote, node 1 is considered as the monitoring node which collects information of all participating nodes in the given topology as shown in figure 5.8.3and figure 5.8.4.

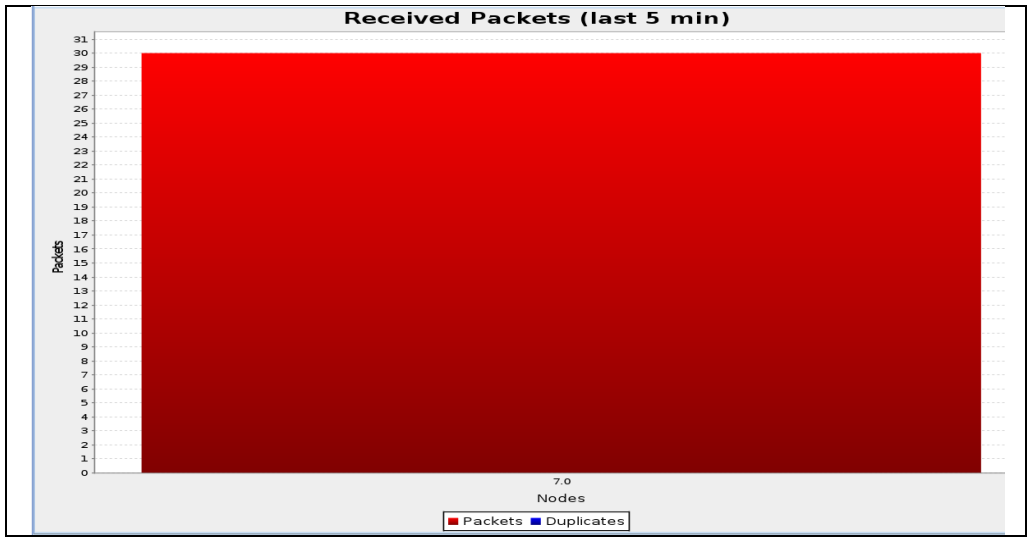


Figure 5.8.1 Received Packets

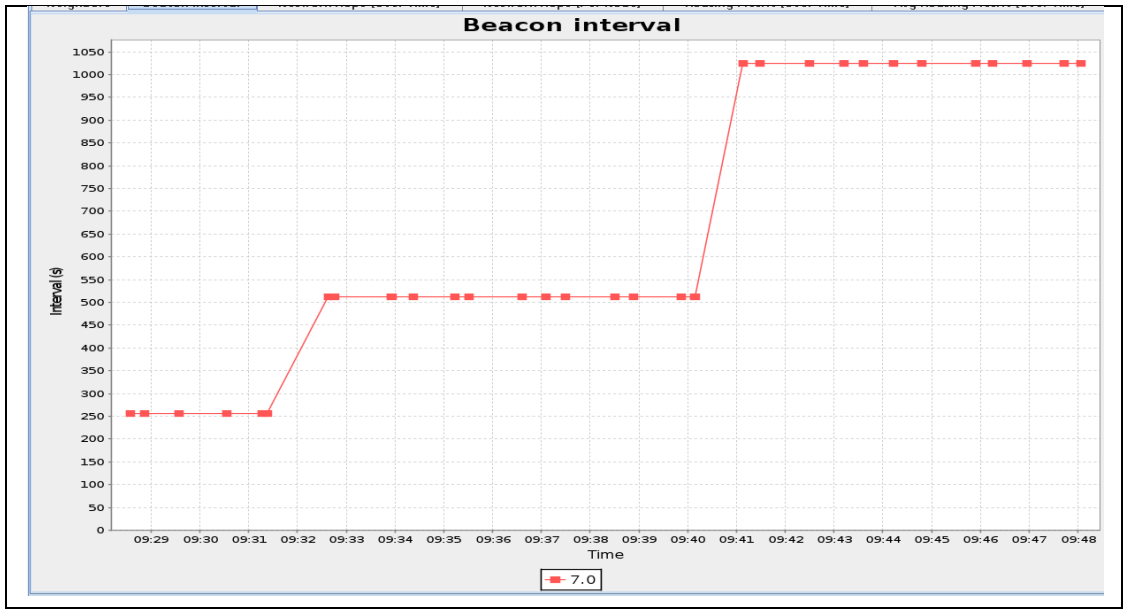


Figure 5.8.2 Beacon Interval

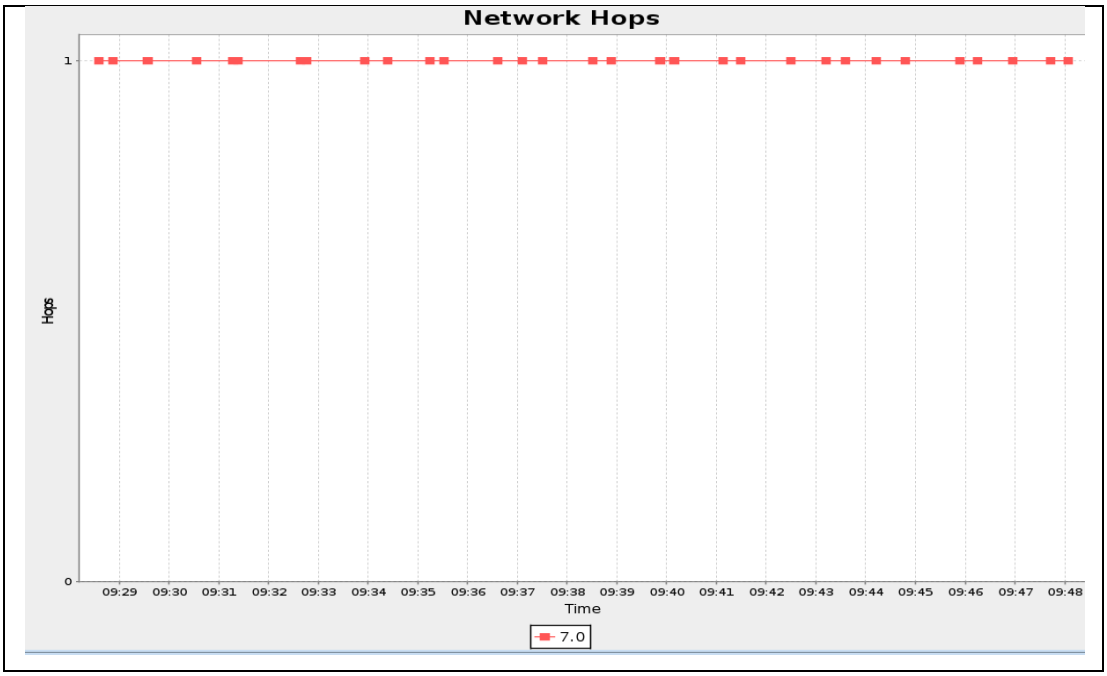


Figure 5.8.3 Network Hops

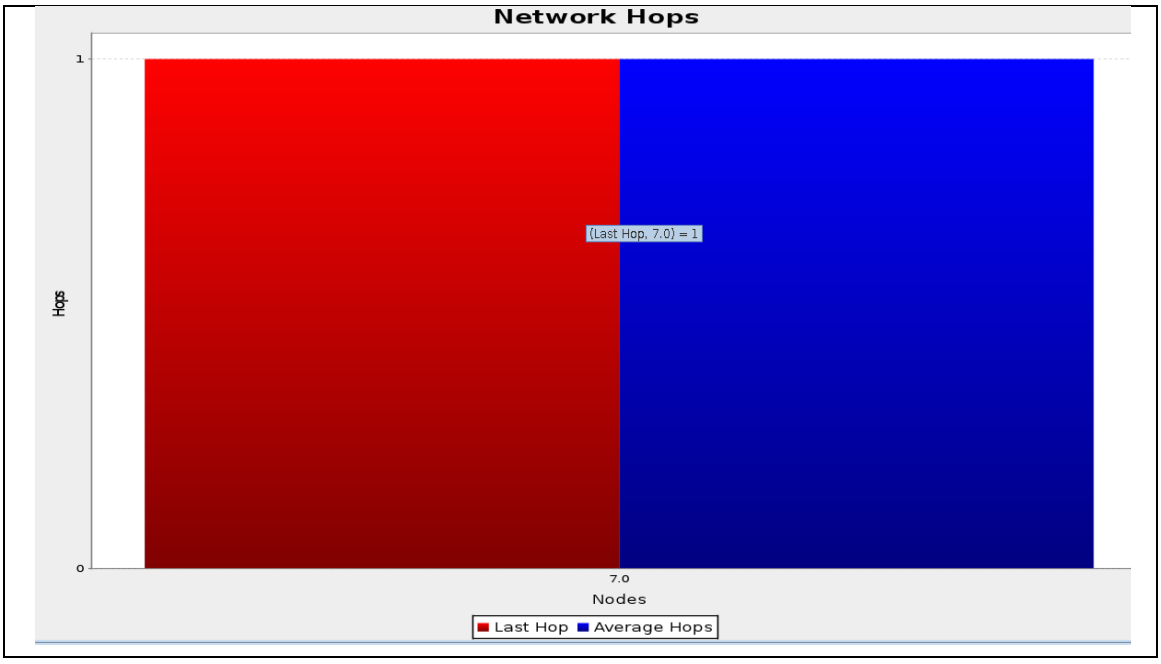


Figure 5.8.4 Network Hops

Figure 5.9.1shows the illustration of routing metric of packet forwarding from the node 8 to rest of the nodes such as 5, 6 and 7. Figure 5.9.2 shows the average routing metric value found to be 32 packets from node 1. Even though node 1 is receiving 32 packets for a stipulated time period, there found to be deviations in the ranges the packet reaches the routing metric value 16. It is found in the illustration that when the value drops from 16 to

value 8, the range stabilizes for certain time period and there is a sudden increase in the deviations between 8 to 16 while it is increasing and 16 to 8 when the value decreases. Routing maintains at the same level till 9:47th second.

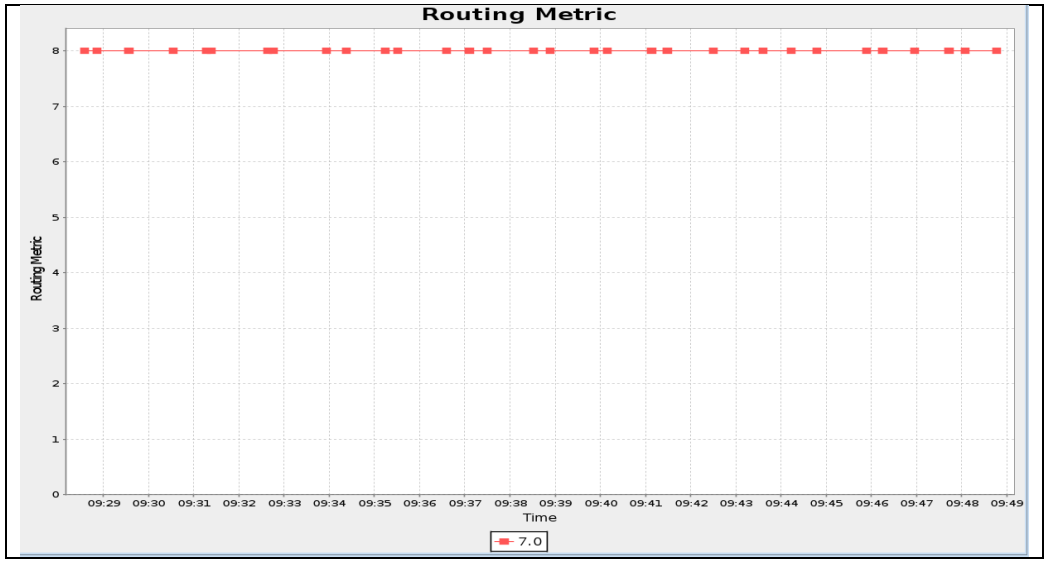


Figure 5.9.1 Routing Metric

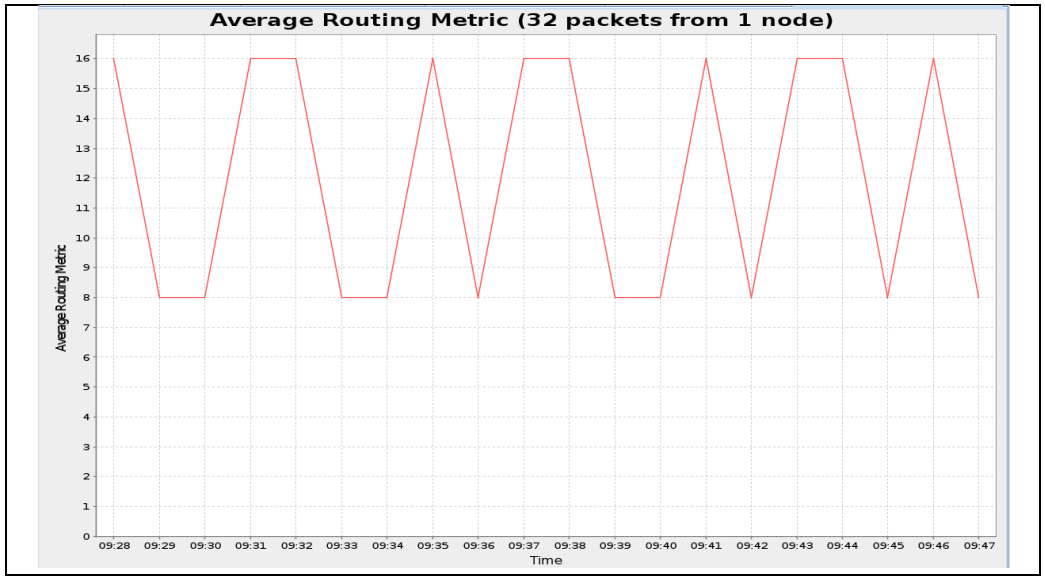


Figure 5.9.2 Average Routing Metric

Figure 5.9.3 shows the average power consumption of participating nodes and power consumption of node 7 is illustrated below in which LPM metric ranges from 0.00 to 0.15, CPU metric ranges from 0.15 to 0.50, radio listen metric ranges from 0.50 to 0.90 and finally the radio transmit value ranges from 0.90 to 0.95. Figure 5.9.4shows the average radio duty cycle in which the radio listens metric ranges from 0.000 to 0.625 and radio transmit value ranges from 0.625 to 0.750. Figure 5.9.5shows the instantaneous power consumption of nodes transmitting data measured in terms of megawatt, in which the Low Power Module value ranges from 0.00mw to 0.15mw, processing metric ranges between 0.15 to 0.48mw, radio listen value ranges from 0.48mw to 0.85mw.

Figure 5.9.6shows the historical power consumption of participating nodes ranges from 0.97mw to 1.11mw and the traffic is observed for the duration of 10 minutes until it reaches 0.98mw.

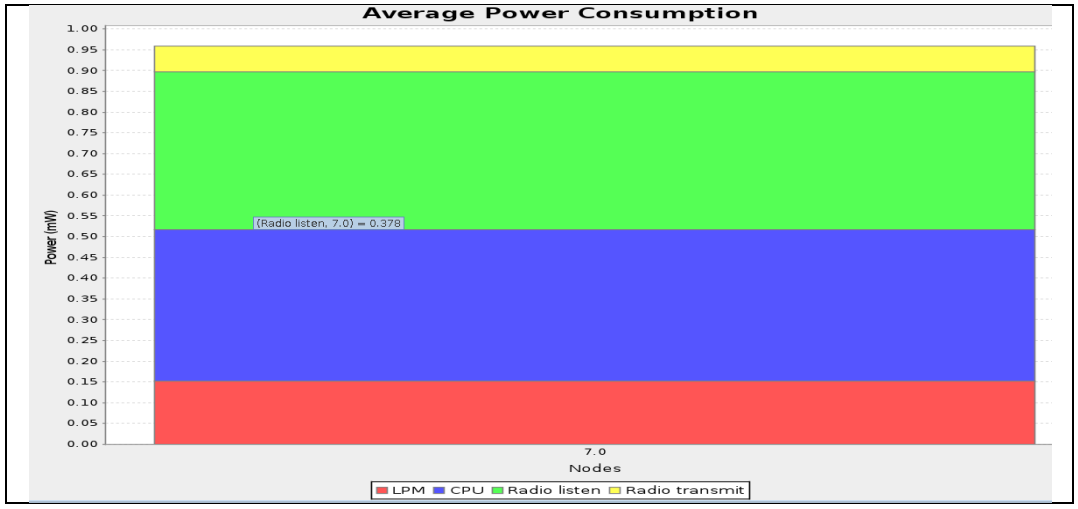


Figure 5.9.3: Average Power Consumption

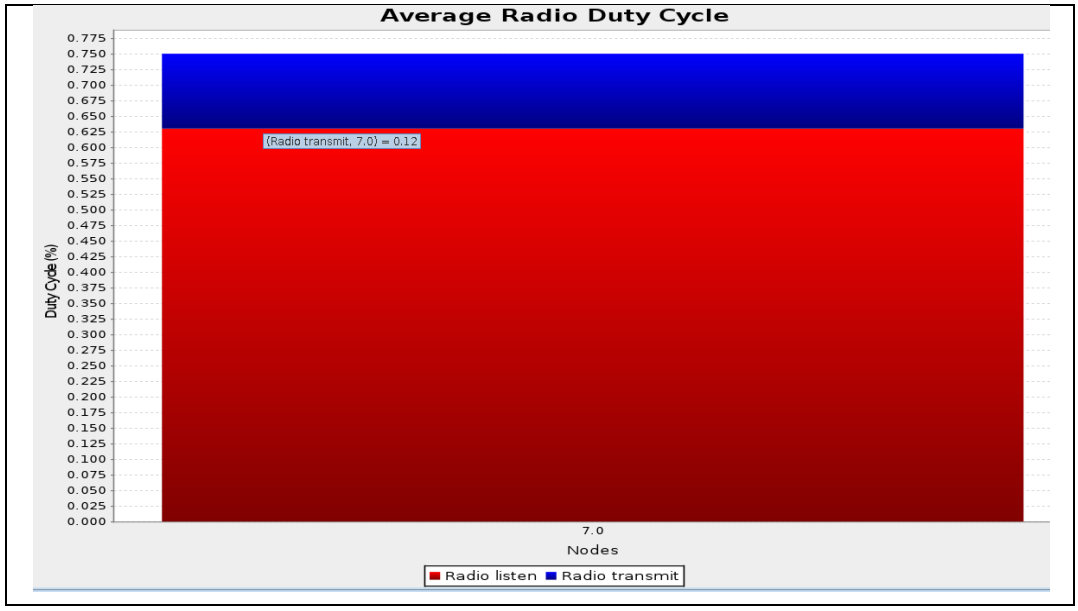


Figure 5.9.4: Average Radio Duty Cycle

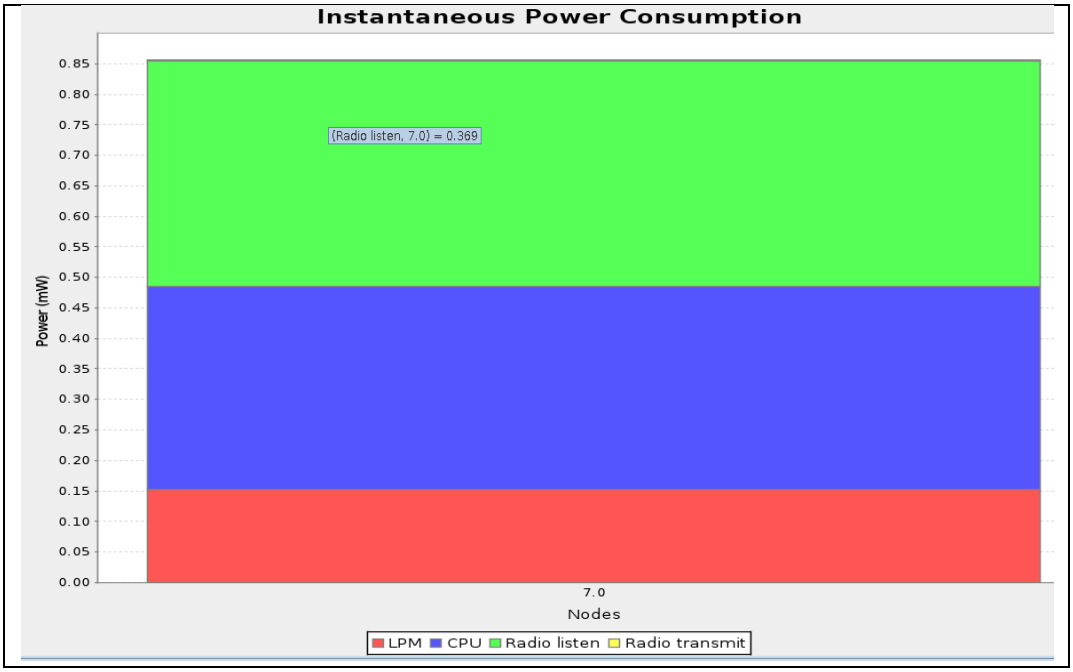


Figure 5.9.5: Instantaneous Power Consumption

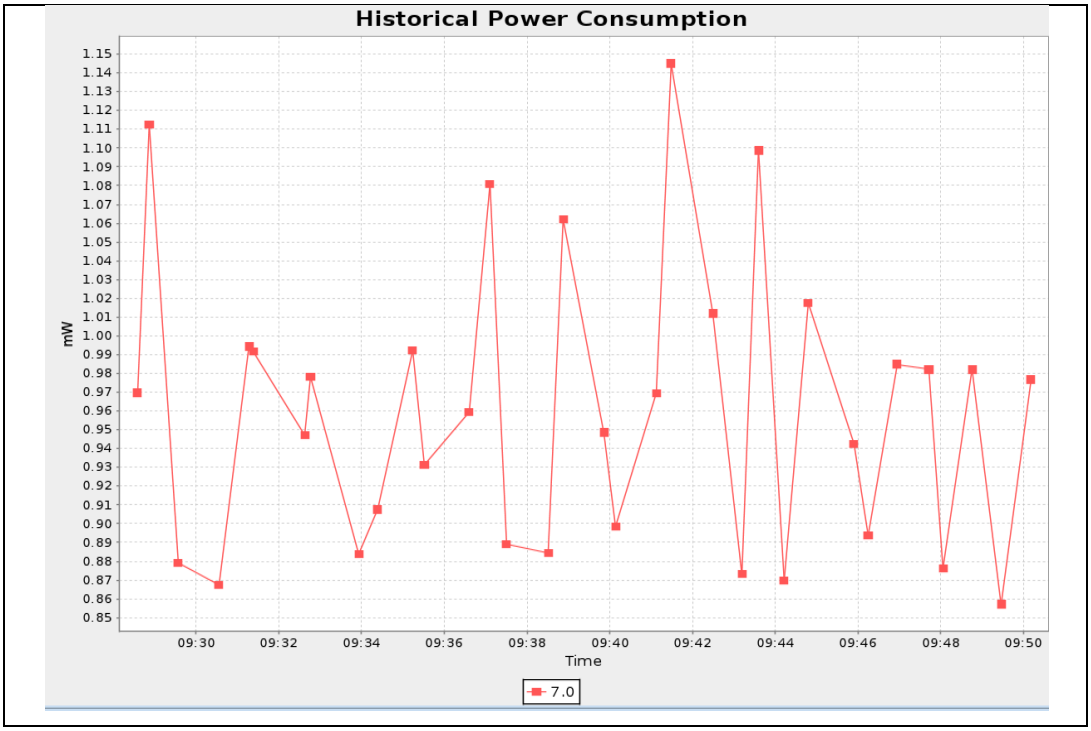


Figure 5.9.6: Historical Power Consumption

Table 5.1.3 and 5.1.4shows the sensor node information deployed on the mote such as participating nodes, received packets, number of hops, routing metric, beacon interval, CPU power, LPM power, Listen Power, Transmit Power, Power consumption, Listen Duty Cycle and Transmit Duty Cycle.

Table 5.1.3 : Details of Wireless Node 5 and 7

Node	**Received Packets**	**Hops**	**Rtmetric**	**Beacon Interval**	**CPU Power**
5.0	0	0.000	0.000	0	0.000
7.0	34	1.000	8.000	11 min, 32 sec	0.365
Avg	**34**	**1.000**	**8.000**	**11 min, 32 sec**	**0.365**

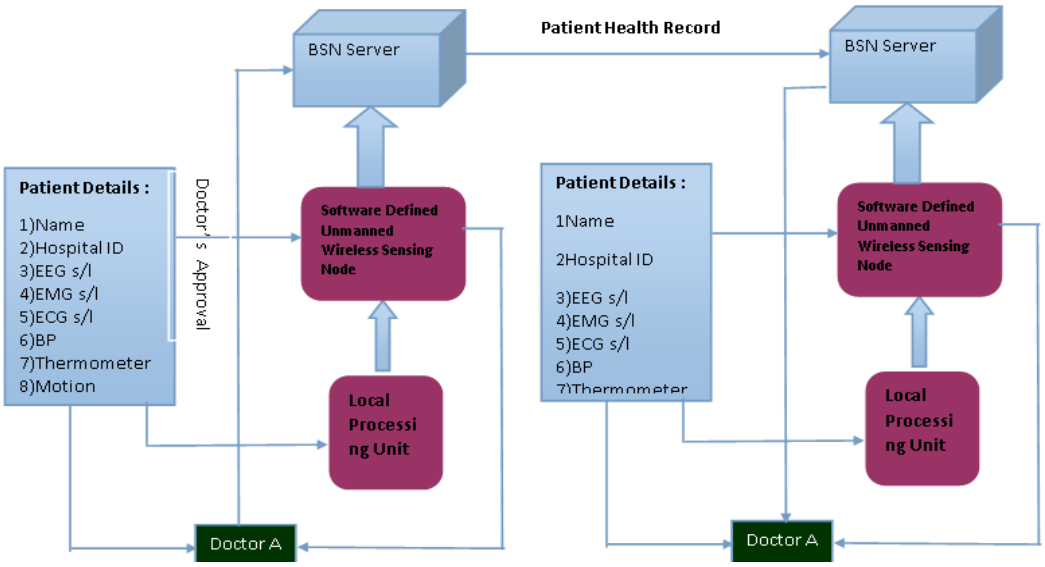
Table 5.1.4 : Details of Other Parameters of Wireless Node 5 and 7

LPM Power	**Listen Power**	**Transmit Power**	**Power**	**Listen Duty Cycle**	**Transmit Duty Cycle**
0.000	0.000	0.000	0.000	0.000	0.000
0.152	0.378	0.065	0.960	0.630	0.122
0.152	**0.378**	**0.065**	**0.960**	**0.630**	**0.122**

VI. Real time applicability of hydiag transposition cipher technique in smart health care system

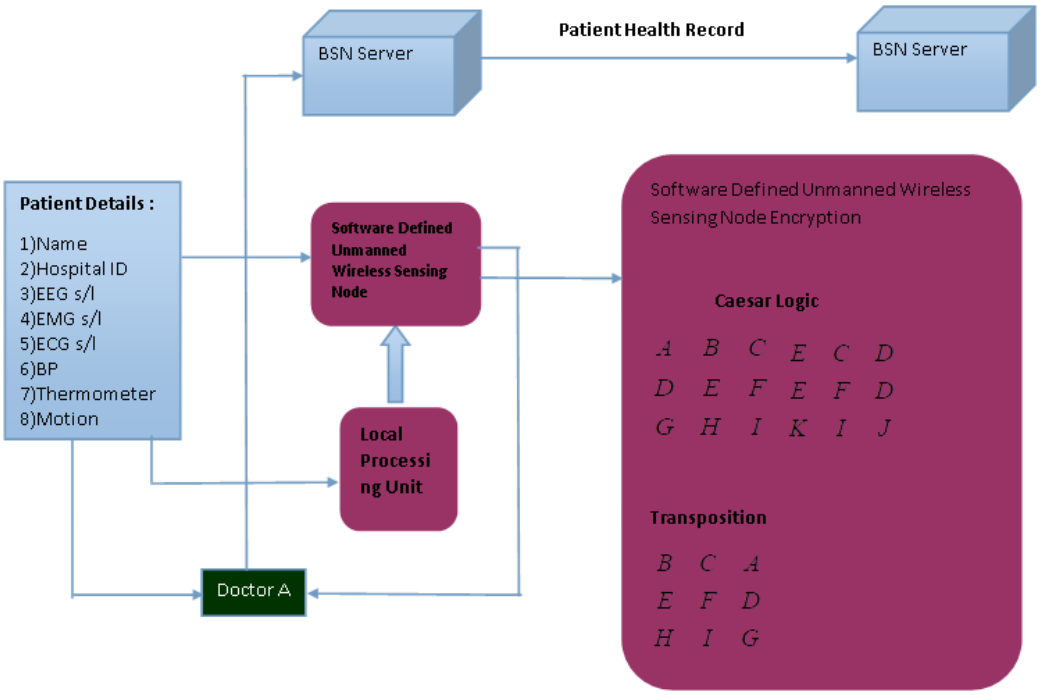
This section discusses about smart health care system which is incorporated with HyDiag transposition cipher model for encrypting and decrypting the patient's diagnosis parameters. The processed data from Local Processing Unit (**LPU**) is sent to the Access Point of the hospital, in turn the data sent to the Body Sensor Network (**BSN**) server in which HYDIAG encryption algorithm is deployed for encrypting the patient's health condition, it is used by the doctor for patient health analysis, patient data is encrypted by making use of HYDIAG algorithm. In critical cases, if doctors are in need to consult with their fellow specialist, encrypted details of a patient will be sent from BSN. As patient details are encrypted at **Hospital network 1**, it is nowhere resulted with the data leakage attacks, Man-in-The-Middle (MITM) attacks over the transmitting channel and the doctors at the other end say **Hospital network 2**, needs to decrypt the enciphered text for obtaining the patient information. The same logic can be applied in smart health care systems for the scenario of high critical operations when doctors wish to discuss with the chief surgeon who is remotely located which needs higher confidentiality for the patient information. Figure 6.1 shows the block diagram of smart health care

system which is deployed with HyDiag transposition cipher model. Figure 6.2 shows the HyDiag Encryption Model at Hospital Network 1 and HyDiag Decryption Model at Hospital Network 2 is shown in Figure 6.3.



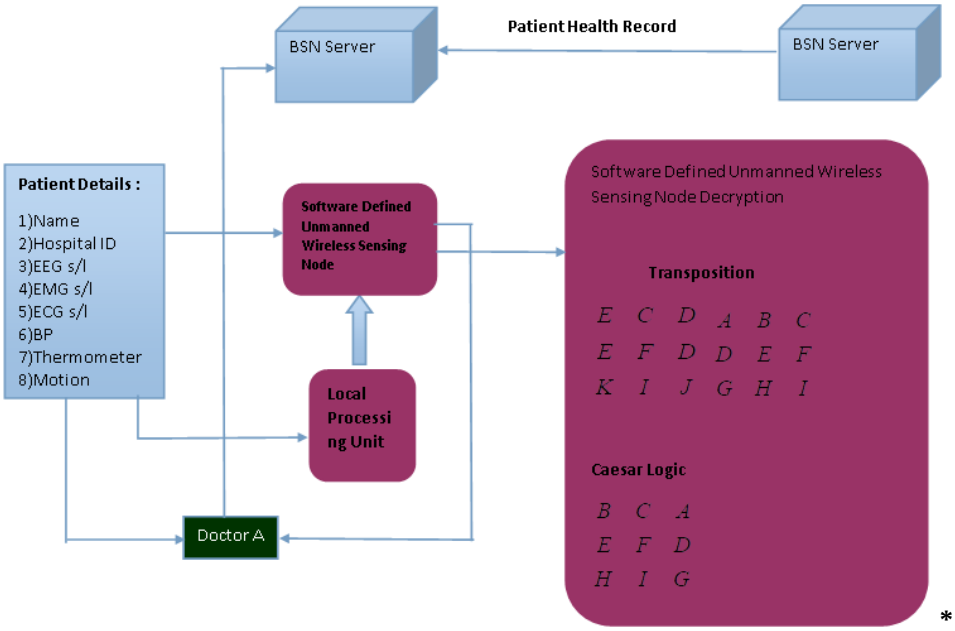
HOSPITAL NETWORK 1 **HOSPITAL NETWORK 2**

Figure 6.1: HyDiag Transposition Cipher Technique in Smart Health Care System



Hospital Network 1

Figure 6.2: HyDiag Encryption Model at the Hospital Network 1



Hospital Network 2

Figure 6.3: HyDiag Decryption Model at the Hospital Network 2

VII. Conclusion and Future Work

In this research work, novel HyDiag cryptographic algorithm is proposed in order to secure confidential data. The proposed HyDiag algorithm is deployed on software defined wireless sensing node which encrypts the messages communicated between data-control and application planes. In case of the occurrence of massive attack traffic flows, software defined wireless sensing node reconfigures security-based alerts to the data plane switches.This research work is not compared with existing methods rather results obtained are observed and interpreted. This research work will be compared with existing methods in future, as of now the authors are initiated to propose novel encryption algorithm. HyDiag transposition cipher model is incorporated with the real time application of smart health care systems to secure the patients information when it is transmitting over to and from the body sensor networks. The algorithm can even be applicable for smart transportation systems to regulate the traffic conditions which is not hackable by attackers, smart agriculture to monitor the crop health condition, to regulate water flow for the crops by transforming the details of actual crop health condition.

References

[1]. Liu Y, Zhao B, Zhao P, Fan P, Liu H. A survey: Typical security issues of software-defined networking. China Communications. 2019 Jul 19;16(7):13-31.
[2]. Pandey, D., Rawat, U., Rathore, N. K., Pandey, K., & Shukla, P. K. (2020). Distributed Biomedical Scheme for Controlled Recovery of Medical Encrypted Images. *IRBM*.
[3]. Cantelli-Forti, A., Capria, A., Saverino, A. L., Berizzi, F., Adami, D., & Callegari, C. (2020). Critical Infrastructure Protection System Design based on SCOUT Multitech SeCurity system for intercOnnected space control groUnd staTions. *International Journal of Critical Infrastructure Protection*, 100407.
[4]. Misuri, Alessio, Nima Khakzad, Genserik Reniers, and Valerio Cozzani. "A Bayesian network methodology for optimal security management of critical infrastructures." *Reliability Engineering & System Safety* 191 (2019): 106112.

[5]. Shameli-Sendi, A. (2020). An efficient security data-driven approach for implementing risk assessment. *Journal of Information Security and Applications*, *54*, 102593.
[6]. Maurya, S., Jain, V. K., & Chowdhury, D. R. (2019). Delay aware energy efficient reliable routing for data transmission in heterogeneous mobile sink wireless sensor network. *Journal of Network and Computer Applications*, *144*, 118-137.
[7]. M. Jammal, T. Singh, A. Shami, "Software Defined Networking: State of art and research challenges," *Computer Networks*, vol. 72, pp.74–98, 2014.
[8]. B.A.A. Nunes, M. Mendonca, X. Nguyen, "A Survey of software-defined networking: Past, present and future of programmable networks, *IEEE Communication Surveys and Tutorial*, vol. 16, No. 3, pp. 1617–1634, 2014.
[9]. H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, 2, pp.114-119, 2013.
[10]. Zhang, Xiaoquan, Lin Cui, Kaimin Wei, Fung Po Tso, Yangyang Ji, and Weijia Jia. "A survey on stateful data plane in software defined networks." *Computer Networks* (2020): 107597.
[11]. Killi, Bala Prakasa Rao, and Seela Veerabhadreswara Rao. "Controller placement in software defined networks: A comprehensive survey." *Computer Networks* 163 (2019): 106883.
[12]. Nisar, Kasif, Ian Welch, Rosilah Hassan, Ali Hassan Sodhro, and Sandeep Pirbhulal. "A Survey on the Architecture, Application, and Security of Software Defined Networking." *Internet of Things* (2020): 100289.
[13]. Javed, U., Iqbal, A., Saleh, S., Haider, S.A. and Ilyas, M.U., 2017. A stochastic model for transit latency in OpenFlow SDNs. *Computer Networks*, *113*, pp.218-229.
[14]. Sviridov, G., Bonola, M., Tulumello, A., Giaccone, P., Bianco, A. and Bianchi, G., 2020. LOcAl DEcisions on Replicated States (LOADER) in programmable dataplanes: Programming abstraction and experimental evaluation. *Computer Networks*, pp.107637.
[15]. Xie, J., Guo, D., Hu, Z., Qu, T. and Lv, P., 2015. Control plane of software defined networks: A survey. *Computer communications*, *67*, pp.1-10.
[16]. Ujcich, B.E. and Sanders, W.H., 2019, June. Data protection intents for software-defined networking. In *2019 IEEE Conference on Network Softwarization (NetSoft)* (pp. 271-275). IEEE.
[17]. Mostovich, D., Fabrikantov, P., Vladyko, A. and Buinevich, M., 2017, February. High-level vulnerabilities of software-defined networking in the context of telecommunication network evolution. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* (pp. 184-186). IEEE.
[18]. Pradhan, A. and Mathew, R., 2020. Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN). *Procedia Computer Science*, *171*, pp.2581-2589.
[19]. Anand, N., Babu, S. and Manoj, B.S., 2018. On detecting compromised controller in software defined networks. *Computer Networks*, *137*, pp.107-118.
[20]. Jagadeesan, L.J. and Mendiratta, V., 2016, October. Programming the network: Application software faults in software-defined networks. In *2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 125-131). IEEE.
[21]. Anadiotis, A.C., Galluccio, L., Milardo, S., Morabito, G. and Palazzo, S., 2019. SD-WISE: a software-defined wireless sensor network. *Computer Networks*, *159*, pp.84-95.
[22]. Almohaimeed and A. Asaduzzaman, "A Novel Moving Target Defense Technique to Secure Communication Links in Software-Defined Networks", IEEE Fifth Conference on Mobile and Secure Services (MobiSecServ), March 2019, pp. 1-4.
[23]. Priya, P.M., 2019, July. Secure Defense Mechanism against Data Leakage and Distributed Denial of Service Attacks in Software Defined Networks. In *2019 International Conference on Communication and Electronics Systems (ICCES)* (pp. 378-382). IEEE.
[24]. MohanaPriya, P. and Shalinie, S.M., 2017. Restricted Boltzmann machine-based cognitive protocol for secure routing in software defined wireless networks. *IET Networks*, *6*(6), pp.162-168.
[25]. Naik, K.P. and Joshi, U.R., 2017, July. Performance analysis of constrained application protocol using Cooja simulator in Contiki OS. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)* (pp. 547-550). IEEE.