ENHANCE THE PERFORMANCE OF WIRELESS SENSOR NETWORK USING ATTACK DETECTION SYSTEM

A PROJECT REPORT

Submitted by

NARMIDHA M (810020106058)

PARKAVI P (810020106061)

REVATHY R (810020106075)

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATON ENGINEERING

UNIVERSITY COLLEGE OF ENGINERING,

BITCAMPUSANNA UNIVERSITY,

TIRUCHIRAPALLI 620024.

ANNA UNIVERSITY,CHENNAI-600025

MAY 2024

ENHANCE THE PERFORMANCE OF WIRELESS SENSOR NETWORK USING ATTACK DETECTION SYSTEM

A PROJECT REPORT

Submitted by

NARMIDHA M (810020106058)

PARKAVI P (810020106061)

REVATHY R (810020106075)

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATON ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING,

BIT CAMPUS, ANNA UNIVERSITY,

TIRUCHIRAPALLI 620024.

ANNA UNIVERSITY,CHENNAI-600025

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project report**, "ENHANCE THE PERFORMANCE OF WIRELESS SENSOR NETWORK USING ATTACK DETECTION SYSTEM"** is the bonafide work of **"NARMIDHA.M, PARKAVI.P, REVATHY.B"** who carried out the project work under my supervision

SIGNATURE

Dr. P. RAMADEVI,

ASSOCIATE PROFESSOR,

HEAD OF THE DEPARTMENT,

Dept. of ECE,

UCE, BIT Campus,

Anna University,

Tiruchirappalli 24

SIGNATURE

Dr. S. G. SUSILA,

SUPERVISOR,

TEACHING FELLOW,

Dept. of ECE,

UCE, BIT Campus,

Anna University,

Tiruchirappalli 24

Submitted to University Viva-Voce examination held on…….....

INTERNAL EXAMINER **EXTERNAL EXAMI**

DECLARATION

We hereby declare that the work entitled "**ENHANCE THE PERFORMANCE OF WIRELESS SENSOR NETWORK USING ATTACK DETECTION SYSTEM''** is submitted for the award of the degree in B.E. Electronics and Communication Engineering, University College of Engineering, BIT Campus, Tiruchirappalli, is a record of our own work carried out by us during the academic year 2023-2024 under the supervision and guidance of **Dr S.G.SUSILA** , Teaching Fellow,Department of ECE, University College of Engineering, BIT Campus, Tiruchirappalli. The extent and the source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other degree or diploma, either in this or other University.

(Signature of the candidate)
NARMIDHA M

(Signature of the candidate)
PARKAVI.P

(Signature of the candidate)
REVATHY.B

I certify that the declaration made by the above candidate is true.

(SIGNATURE OF THE SUPERVISOR)
Dr. S. G. SUSILA,
Teaching Fellow
Department of Electronics and Communication Engineering,
University College Of Engineering, BIT Campus,
Anna University, Trichirappalli.

ABSTRACT

Communication in cyber-physical systems relies heavily on Wireless Sensor Networks (WSNs), which have numerous uses including ambient monitoring, object recognition, and data transmission. A WSN consists of a large number of sensor nodes with limited batteries; the sensoring devices are deployed randomly on a zone to collect data. WSNs are threatened by several malicious behaviors caused by some nodes. The impact of such behaviors can be serious, even fatal, due to the collaborative nature of nodes in a network without fixed infrastructure. Then, in this work we have referred to the need for a secure network communication network with mechanisms that take into account the limited resources of the nodes. In order to achieve such security, the network can be split into sectors, and mobile agents (MAs) can be used to reject traffic intruders caused by Wormhole attacks taking into account energy constraint. Wormhole attack is a Denial of service attack launched by malicious nodes by creating a tunnel through which the packets are replayed to malicious nodes disrupting the communication channel and corrupting network routing. In order to evaluate the performance of our proposal, we have carried out several simulation tests using the NS2 simulator. Hence our proposal extends the life of the network, in terms of energy consumption and the rate of packet delivery.

ABSTRACT

சைபர்-இயற்பியல் அமைப்புகளில் தகவல் தொடர்பு வயர்லெஸ் சென்சார் நெட்வொர்க்குகளை (WSN கள்) பெரிதும் நம்பியுள்ளது, அவை சுற்றுப்புற கண்காணிப்பு, பொருள் அங்கீகாரம் மற்றும் தரவு பரிமாற்றம் உள்ளிட்ட பல பயன்பாடுகளைக் கொண்டுள்ளன. ஒரு WSN ஆனது வரையறுக்கப்பட்ட பேட்டரிகளைக் கொண்ட அதிக எண்ணிக்கையிலான சென்சார் முனைகளைக் கொண்டுள்ளது; உணர்திறன் சாதனங்கள் தரவைச் சேகரிக்க ஒரு மண்டலத்தில் தோராயமாக பயன்படுத்தப்படுகின்றன. சில முனைகளால் ஏற்படும் பல தீங்கிழைக்கும் நடத்தைகளால் WSN கள் அச்சுறுத்தப்படுகின்றன. நிலையான உள்கட்டமைப்பு இல்லாத நெட்வொர்க்கில் உள்ள முனைகளின் கூட்டுத் தன்மை காரணமாக இத்தகைய நடத்தைகளின் தாக்கம் தீவிரமானதாகவும், ஆபத்தானதாகவும் இருக்கலாம். பின்னர், இந்த வேலையில் முனைகளின் வரையறுக்கப்பட்ட வளங்களை கணக்கில் எடுத்துக் கொள்ளும் வழிமுறைகளுடன் பாதுகாப்பான நெட்வொர்க் தகவல்தொடர்பு நெட்வொர்க்கின் அவசியத்தை நாங்கள் குறிப்பிட்டுள்ளோம். அத்தகைய பாதுகாப்பை அடைவதற்காக, நெட்வொர்க்கை பிரிவுகளாகப் பிரிக்கலாம், மேலும் ஆற்றல் தடையை கணக்கில் எடுத்துக்கொண்டு வோர்ம்ஹோல் தாக்குதல்களால் ஏற்படும் போக்குவரத்து ஊடுருவல்களை நிராகரிக்க மொபைல் முகவர்கள் (எம். ஏ. க்கள்) பயன்படுத்தப்படலாம். வோர்ம்ஹோல் தாக்குதல் என்பது தீங்கிழைக்கும் முனைகளால் தொடங்கப்பட்ட சேவை மறுப்பு தாக்குதலாகும்,

ACKNOWLEDGEMENT

All praise, glory and honour to the Lord Almighty, for his gracious presence and guidance that enable us to complete this project duly.

We wish to express our profound thanks to **Dr T. SENTHILKUMAR**, Dean, University College of Engineering, BIT Campus, Tiruchirappalli, for granting us permission for doing this project work.

We wish to express our sincere thanks to **Dr . C. RAMADEVI,** Head of the Department of Electronics and Communication Engineering, University College of Engineering, BIT Campus, Tiruchirappalli, for her support and ardent guidance.

We owe our special thanks and gratitude to **Dr S.G.SUSILA** Teaching Fellow, Department of ECE, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli, who guided us throughout our project with her timely help, guidance with valuable suggestion.

We express our heartfelt thanks to our project review committee members, Department of Electronics and Communication Engineering for their passionate support, for helping us to identify our mistakes and also for the appreciation they gave us in achieving our goals. We heartily thank our lab assistants and management for their support by way of providing the information and resources that helped us to complete the project successfully.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	ACRONYMS
1.	WSN	Wireless Sensor Networks
2.	IOT	Internet Of Things
3.	MAC	Medium Access Control
4.	LLC	Logical Link Control
5.	LAN	Local Area Network
6.	ALOA	Additive Link On-Line Hawaii System
7.	CSMA	Carrier Sense Multiple Access
8.	LPL	Low Power Listening
9.	FDM	Frequency Division Multiplexing
10.	TDM	Time Division Multiplexing
11.	CDMA	Code Division Multiple Access
12.	Z-MAC	Zebra MAC
13.	S-MAC	Scheduled Channel Polling MAC
14.	TAS-MAC	Traffic Adaptive Synchronous MAC
15.	CASS	Compressive Adaptive Sense & Search
16.	SNR	Signal to Noise Ratio
17.	SGKP	Secure Group Key Protocol
18.	PKG	Private Key Generator
19.	TCL	Tool Command Language
20.	GVFS	Gnome VFS
21.	RREQ	Route Request
22.	SET	Secure Efficient Transmission
23.	MLU	Maximum Link Utilization

CHAPTER 1

INTRODUCTION

Utilizing scattered Wireless Sensor Networks (WSN) for gathering environmental information from a large number of sensor nodes with limited capabilities continues to draw a lot of attention from both industrial and academic communities, due to the large number of applications that rely on such infrastructures. For example, WSN will be a crucial technology enabler for implementation in the emerging Internet of Things (IOT) environment, which will allow collection of information from densely deployed sensors, many of which are cheap with very limited capabilities (e.g., memory or energy resources) and compete for limited wireless resources (e.g., time or spectrum). In many of these applications, the majority of the sensors need to report only a limited amount of information, and do so only infrequently. For example, many of these sensors need to periodically send a keep alive message to inform the sink node that their battery has not drained out, and occasionally report one of several possible events, e.g., motion detected, temperature is above a given threshold, or one out of several quantized values. Accordingly, the main challenge for such networks is to cope with a huge number of simple devices that need to send limited information, competing for very limited wireless resources (compared to the number of sensors) while saving energy.

Here Improvements in hardware technology have resulted in low-cost sensor nodes, which are composed of single chip embedded with memory, processor, and transceiver. Low-power capacities lead to limited coverage and communication range for sensor nodes compared to other mobile devices. Hence, for example, in target tracking and border surveillance applications, sensor networks must include

a large number of nodes in order to cover the target area successfully.

Unlike other wireless networks, it is generally difficult or impractical to charge/replace exhausted batteries. That is why the primary objective in wireless sensor networks design is maximizing node/network lifetime, leaving the other performance metrics as secondary objectives. Since the communication of sensor nodes will be more energy consuming than their computation, it is a primary concern to minimize communication while achieving the desired network operation.

1.1 SCOPE OF THE PROJECT

MAC-LAYER-RELATED SENSOR NETWORK PROPERTIES

The MAC sub-layer is a part of the data link layer specified in the communication protocol stack and is shown in Figure 1. It provides the channel access mechanism to several medium sharing devices. On a wireless medium, which is shared by multiple devices and is broadcast in nature, when one device transmits, every other device in the transmission range receives its transmission. This could lead to an interference and collision of the frames when a transmission from two or more devices arrives at one point simultaneously. Sensor nodes usually communicate via multi-hop paths over the wireless medium in a scattered, dense, and rough sensor field. A MAC protocol manages the communication traffic on a shared medium and creates a basic network infrastructure for sensor nodes to communicate with each other. Thus it provides a self-organizing capability to nodes and tries to enforce the singularity in the network by letting the sender and receiver communicate with each other in a collision and error-free fashion.

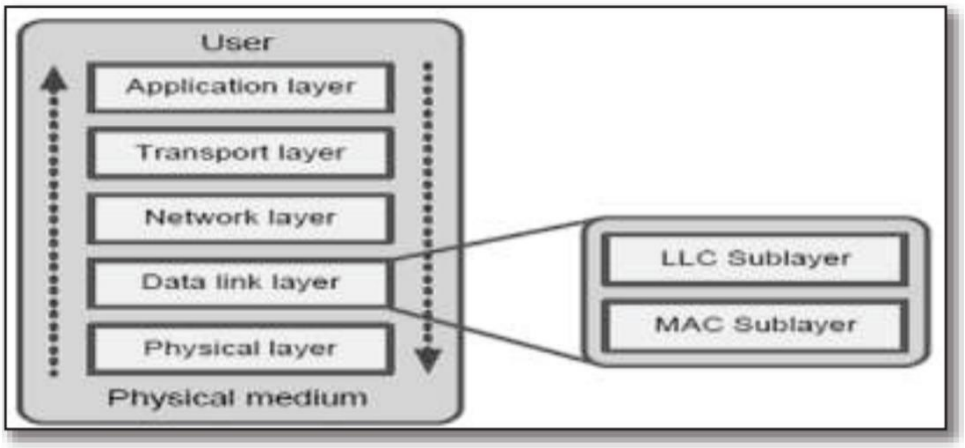


Fig. 1.1. The Communication Protocol Stack

This five-layered simplified model is commonly applied to network research as apposite to the seven-layered OSI model. An end-user can use application specific software/algorithms at the application layer. The transport layer helps maintaining the sensor data flow. The network layer routes data on an appropriate path. The LLC sub-layer of the data link layer provides framing, flow control, error control, and link management facilities, whereas the MAC sub-layer manages collisions and helps in energy aware operations of sensor nodes. The physical layer takes care of the radio, channel, modulation, transmission, and reception of bits on a physical medium.

Moreover, the typical requirements to increase lifetime of a WSN without the need of any power replacement and/or human interaction has prompted the development of novel protocols in all layers of the communication stack. However, prime gains can be achieved at the data link layer, where the MAC protocol directly controls the activities of the radio, which is the most power consuming component of resource-scarce sensor nodes. Efficient MAC protocols utilize the radio judiciously to conserve its energy. Thus the MAC protocol helps fulfilling important design objectives of WSNs by specifying how nodes employ the radio, share the channel, avoid collision in correlated and broadcasting environments,

response the inquirer timely, and survive for a longer period. Hence, designing novel solutions for MAC protocols for WSNs has been and will remain a focal point for many researchers.

At the same time, a MAC protocol can be made accountable for the following sources of energy waste in WSNs, which mainly relate to the communication.

- **Idle listening**: Since a node in a WSN usually does not know when it will be the receiver of a message, it keeps its radio in ready-to-receive mode, which consumes almost as much energy as in receive mode. In low traffic applications, this is considered one of the major sources of energy waste. Note that carrier sensing, which a MAC protocol requires to sense the current status of the channel, is not a part of idle listening.
- **Collisions**: A collision is a wasted effort when two frames collide with each other and are discarded because the receiver has to drop the overlapped information. A collision usually results in retransmission and drains more energy in transmitting and receiving extra packets. The half-duplex nature of the wireless medium precludes collision detection, thereby increasing the responsibilities of the MAC protocol. The high density of the deployed nodes, on one hand, helps improving network connectivity without compromising the transmission power. However, on the other hand, it increases collision probability for the MAC protocol by increasing the number of nodes contending for the channel.
- **Overhearing**: An overhearing occurs on the wireless broadcast medium when the node receives and processes a gratuitous packet that is not addressed to it. In the dense network and under heavy traffic situations, this could lead to a serious problem.

- **Control packet overhead**: An increase in the number and size of control packets results in overhead and unnecessary energy waste for WSNs, especially when only a few bytes of real data are transmitted in each message. Such control signals also decrease the channel capacity. A balanced approach is required so that the required number of control packets can be kept at minimal.
- **Over-emitting**: An over-emitting or a deafness occurs due to the transmission of the message when the destination node is not ready to receive it.
- **Complexity**: Computationally expensive algorithms might decrease the time the node spends in the sleep mode. They might limit the processing time available for the application and other functionalities of the protocol. An overly simple MAC algorithm can save higher energy than a complex one, but it may not be able to provide the complex functions such as adaptation to traffic and topology conditions, clustering, or data aggregation.

1.2 OBJECTIVE OF THE PROJECT

This project is to evaluate the performance of WSN by Attack detection system through simulation tests using NS2 simulator. The Performance are Energy consumption,Throughput,Delay.

WSN-MAC PROTOCOL

WSN MAC protocols were widely investigated during the last decades. In this section we discuss the main results for MAC and for cross-layer protocols which developed to enhance energy and network lifetime. As a summary of the individual characteristics of the protocols mentioned in the next sub-section, table I

at the end represents comparative summary.

Medium Access Control Protocols

Medium Access Control (MAC) protocol has a frame format which is used to provide the data link layer of the Ethernet LAN system to control access over the communication channel. In general, well-known MAC protocols include Ethernet and MAC is used in the IEEE 802.11 (Wi-Fi) family. However, there are several various types of MACs have been developed for WSNs.

CHANNEL ACCESSING CHRONOLOGY

The nature collision in wireless broadcast medium requires an efficient channel accessing method to control access to the shard medium. Therefore, this collision can be offer free communication among nodes. Specifically, accessing the channel is classified into two major categorizations; contention based networks and contention free networks. In contention based networks, devices are contending each other to gain access of the channel. Whereby, contention free networks uses time or frequency to schedule the channel. In this category, devices can only access their allocated channel slots, and these devices communicate with the central node in a collision free method.

In the other hand, accessing channel scenarios have been already proposed to find the answer of who is allowed to access and how can access. However, the Additive Link On-Line Hawaii System (ALOHA) protocol 5 is proposed in 1970s and also defined as pure ALOHA. This protocol is considered as one of the pioneer protocols in this category. It allows devices data to be transmitted immediately when they have data to send. ALOHA is a simple and decentralized MAC protocol works seamlessly under low loads, likewise the maximum channel utilization in ALOHA is only 18.4% 5. In ALOHA, the slotted ALOHA is used to double

ALOHA utilization by subdividing the time into slots. In this case, collisions can occur only at the beginning of the slot, since the node is allowed to start a transmission only at the beginning of a slot. However, slotted-ALOHA reduces the collisions probability by doing synchronization among nodes.

Commonly, the Carrier Sense Multiple Access (CSMA) scheme is used in wired and wireless Local Area Networks (LANs). CSMA can capably sense the transmission of other nodes before starting a node its transmission. Therefore, CSMA is considered as a contention based access protocol, as well as it is simple, flexible and robust especially for the dynamic networks topology. However, this scheme is still suffering from serious energy waste, high overhead and throughput degradation on the already resource constrained sensor nodes which are caused by the additional collisions.

Classification of WSN MAC

Protocols Several MAC protocols have been successfully proposed to meet the stringent design requirements of WSNs. Actually these protocols depend on how protocol allows nodes to access the channel. We have classified WSN based MAC protocol as depicted in Figure into four categories as; contention based, scheduling based, channel polling based, and hybrid protocols.

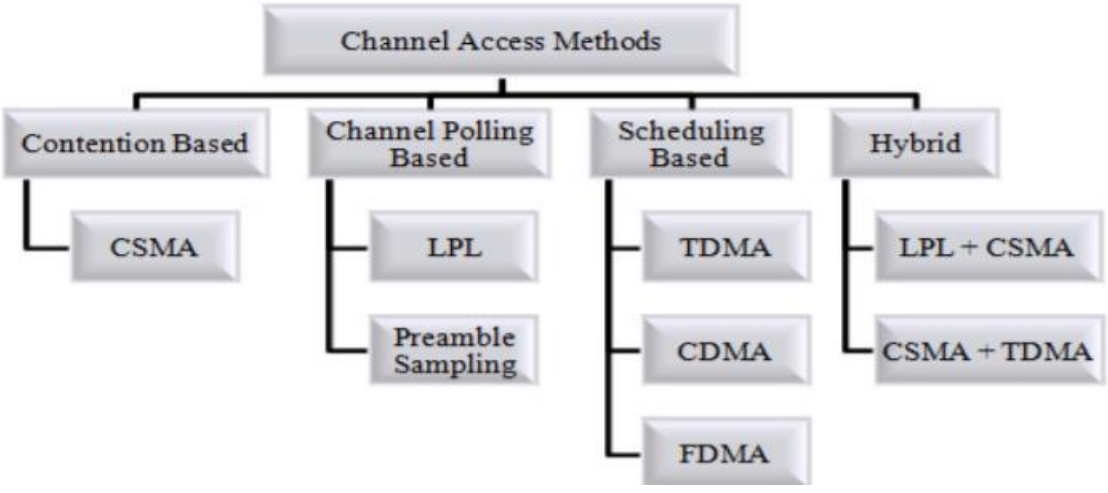


Fig1.2 Channel Accessing Taxonomy in WSNs

Contention Based MAC Protocols

As mentioned earlier, nodes using contention based schemes are working on acquiring the channel. Hence, the network node competes with its neighbours to get the channel. This process will be done when the node senses the carrier before getting start with data transmission. If the carrier is set up as idle, then node will start its transmission, otherwise node will defer the transmission for some time randomly. This deferring is usually determined by a backoff algorithm. Event-driven WSN applications use contention based MAC protocols to reduce the processing resources consumption. However, contention based MAC protocols are flexible and dynamics to network scales.

Channel Polling Based MAC Protocols

Channel polling scheme is known as a preamble sampling and Low Power Listening (LPL). Moreover, sending prefixes data packets with extra bytes by node is called preamble. Specifically, node sends the preamble over the channel to ensure that the destination node detects the radio activity and wakes up before receiving the actual payload from the sender. On a wake-up, if a radio activity is detected by receiver, then the receiver will turn on its radio to receive data packets. Otherwise, the node (receiver) goes back to the sleep mode until the next polling interval.

Scheduling Based MAC Protocols

During the initialization phase, scheduling based schemes assign collision-free links between neighbouring nodes. However, links may be allocated as frequency division multiplexing (FDM) bands, time division multiplexing (TDM) slots, or code division multiple access (CDMA) based spread spectrum codes. Due

to the complexities that incurred with FDMA and CDMA schemes, therefore, WSNs prefer TDMA schemes as scheduling methods to reduce the incurred complexity. In TDMA schemes, the system time is divided into slots. These slots are then assigned to all the neighbouring nodes.

Hybrid MAC Protocols

In order to achieve a joint improvement, hybrid MAC protocols combine the strengths of two or more different MAC schemes. Usually, hybrid MAC protocols combine a synchronized scheme with an asynchronous scheme. Though hybrid protocols cumulative the advantages of multiple schemes, they can also carry, scaling and complexity problems in maintaining two or more different working modes. However, Zebra MAC (Z-MAC) protocol is one of the most important examples in hybrid scheme, which combines the strengths of TDMA and CSMA while offsetting their weaknesses. As well as, the Scheduled Channel Polling MAC (SCP-MAC) and Funnelling-MAC protocol are also two important examples on this scheme.

1.3 REPORT SUMMARY

DATA GATHERING IN WSN

Data gathering is mainly for estimating network size, determining average system load, processing user queries, gathering interesting data, and so on. Much effort has been spent on designing effective data gathering approaches with different focuses such as energy-efficiency, network lifetime, delay bound, network throughput, energy-delay tradeoff, and so on.

In general, data gathering can be further classified as data collection with aggregation and data collection without aggregation, referred to as data aggregation and data collection respectively. In data aggregation, specific

aggregation functions are employed during the data gathering process, e.g., MAX, MIN, SUM, AVERAGE, and so on. In data collection, all the raw data produced at each node is gathered to the sink (base station) without any aggregation function.

Figure 1.3 shows an example WSN consisting of one sink node denoted by s0 and 8 sensor nodes denoted by si ($1 \leq i \leq 8$). s0 can communicate directly with a PC or some other user-operated devices in a wired or wireless manner. Each bidirectional edge between two nodes in Fig. 1.3 implies these two nodes have a one-hop communication link. For two nodes without a direct link, they can communicate via a multi-hop manner as long as the network is connected. For instance, in Fig. 1.3, s1 and s8 can communicate along the path s1 ↔ s4 ↔ s5 ↔ s8 or other paths. Now, suppose the WSN in Fig. 1.3 is used for monitoring the temperature of the deployed area. At a particular time instant t, assume the temperature values sensed by si ($1 \leq i \leq 8$) are 48◦ at s1, 58◦ at s2, 60◦ at s3, 57◦ at s4, 61◦ at s5, 58◦ at s6, 47◦ at s7, and 49◦ at s8, respectively.

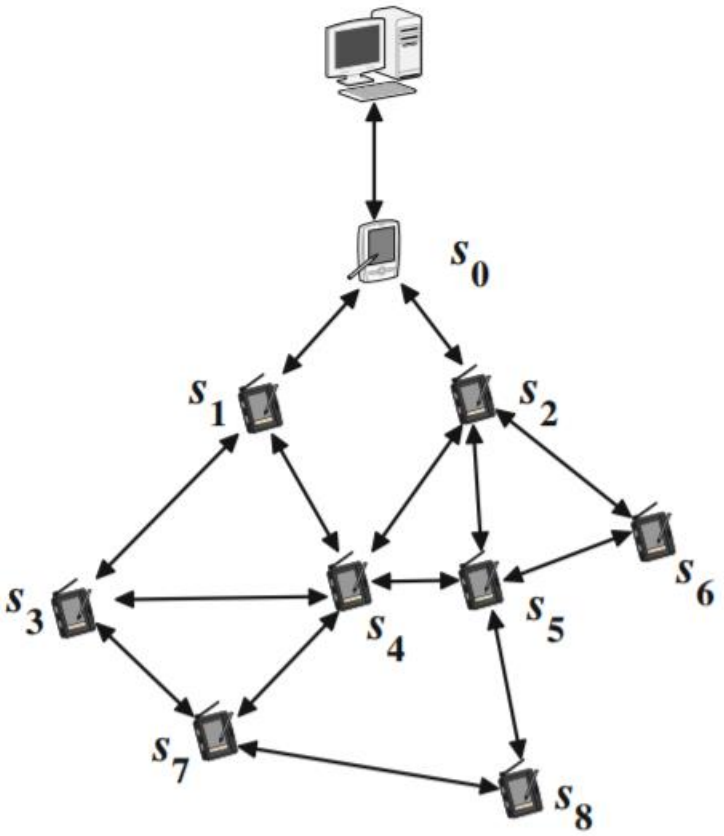


Fig.1.3 WSN and Data Gathering

Then, if we want to obtain the maximum temperature value of the monitored area at time t, we can conduct data aggregation in this network by applying the aggregation function MAX, which is used to obtain the maximum value of a candidate data set. Consequently, in this example, the data aggregation value is MAX{48◦, 58◦, 60◦, 57◦, 61◦, 58◦, 47◦, 49◦} = 61◦. On the other hand, if we want to perform data collection over this network, e.g., to collect all the temperature values sensed by all the sensors at time t to the sink, all the 8 temperature values in the set {48◦, 58◦, 60◦, 57◦, 61◦, 58◦, 47◦, 49◦} will be collected to the sink for further processing.

When dealing with data aggregation and data collection for WSNs, the constraints and limitations of WSNs on power supply, computation ability, and

communication capacity introduce many challenges. We summarize some main challenges as follows.

- **Energy Efficiency:** Most sensor nodes are battery-powered, which implies the available power of a sensor node is very limited. Furthermore, in large-scale WSNs and WSNs which are deployed as monitoring and controlling systems where human intervention is not desirable or feasible, it is usually impossible to recharge a sensor node. Therefore, how to design energy-efficient data aggregation and data collection methods is a challenge. Especially for data collection, more traffic will be induced compared with data aggregation.
- **Timeliness and Real-time:** The limited communication capacity of each sensor node and the interference caused by wireless communication makes it more challenging to design effective data aggregation and collection algorithms for real-time applications. It is worth mentioning that more communication traffic will be induced for data collection compared with data aggregation, especially for the nodes close to the sink. Apparently, data are easily accumulated at the nodes close to the sink, which is called the data accumulation phenomenon. Thus, to resolve this problem and design elegant scheduling schemes is very important.
- **Scalability and Robustness:** WSNs tend to be large-scale networks and distributed systems. Some new nodes may join a network and some existing nodes may disappear at any time. This implies that the topological structure of a WSN varies over time. To deal with the dynamics, distributed algorithms with scalability are highly desired. On the other hand, it is difficult, sometimes impossible, to obtain the overall real-time network information to design

optimal distributed algorithms. Therefore, designing optimal or sub-optimal distributed data aggregation and collection algorithms is challenging.

- **Time Synchronization and Distributed Solutions:** For large-scale WSNs consisting of vulnerable sensor nodes, it might be difficult and not realistic to achieve ideally strict time synchronization due to the unstable deployment environment, clock drifts, and technical limits.1 Therefore, to comprehensively and profoundly understand the performance of data aggregation and collection for practical WSNs, it is also important to investigate distributed data aggregation and collection algorithms for asynchronous WSNs.
- **Other Issues:** There exist many other challenges for data aggregation and collection in WSNs induced by node mobility, duty cycling, security issues, and so on, in different kinds of applications. To address these challenges, application-aware data aggregation and collection schemes are expected.

In particular, traffic in a WSN can be quite dynamic, depending on the events being sensed, the sensing application and the protocols being used. Therefore, such protocols should perform well under a wide range of traffic loads, a variety of network topologies and various objectives such as latency, reliability, energy consumption, security, etc. Accordingly, many of these MAC protocols were designed and examined to be robust under diverse setups. The approach of designing a MAC protocol regardless of the wireless Phy-protocol being used, the routing protocol employed, and even the application expected to utilize it, is consistent with the network-layering conceptual model. However, even though these protocols perform well under a large variety of setups, this universality comes at a price.

For example, a MAC protocol allowing any subset of the sensors, regardless of its size, to transmit simultaneously will suffer from many collisions and high overhead if indeed a large subset attempts transmission concurrently. Another example is a keep alive frame transmitted over WiFi, that conveys only the sender ID and a single bit of relevant information (the sensor is alive), yet requires a large number of bits to be transferred due to headers, physical encapsulation, etc.

In this project, we design, analyze and evaluate a highly efficient WSN MAC protocol specially designed to collect information from a large number of sensors, utilizing information theoretic concepts and novel signaling and decoding techniques which allow us to jointly optimize all layers together. We assume the sensors are very simple, with highly constrained capabilities, e.g., no power control, rate adaptation mechanisms or sophisticated algorithmic capabilities. Thus, the key idea that our protocol relies on, is that instead of the typical frame mechanism using data encapsulation, each sensor is assigned a unique transmission pattern for each of its messages, which conveys both the information and the sensor's ID. Whenever a sensor wishes to transmit a report, it waits to receive a predefined periodic preamble sent by its designated sink, and then transmits a sequence of impulses according to the transmission pattern which corresponds to the report it wishes to send.

The sink node can receive and decode several simultaneous transmissions in a way that it can recognize both the sender and the information sent from the collected channel output received. Namely, a collision resolution procedure or scheduling are not required to decode the K simultaneously transmission. This is done using a carefully designed codebook, and a matching decoding algorithm that identifies both the sensors which transmitted as well as their code words. Interestingly, unlike Code-Division Multiple-Access Channels (CDMA), the sink

node can rely on a simple energy detection in order to decode, and not on the exact received power or any power adaptation mechanism, thus dramatically improving robustness.

Group Key Management Scheme: is a single key which is assigned only for one group of mobile nodes in WSN. For establishing a group key, group key is creating and distributing a secret for group members. There are specifically three categories of group key protocol

1. Centralized, in which the controlling and rekeying of group is being done by one entity.
2. Distributed, group members or a mobile node which comes in group are equally responsible for making the group key, distribute the group key and also for rekeying the group.
3. Decentralized, more than one entity is responsible for making, distributing and rekeying the group key.

Some important Group key Management schemes in WSN are Simple and Efficient Group Key Management (SEGK), and Private Group Signature Key (PGSK). Hybrid Key Management Scheme: Hybrid or composite keys are those key which are made from the combination of two or more than two keys and it may be symmetric or a asymmetric or the combination of symmetric & asymmetric key. Some of the important Hybrid key management schemes in WSN are Cluster Based Composite Key Management, and Zone-Based Key Management Scheme.

Secure selection of cluster heads is another significant issue in WSNs. The general approach in former protocols is that each participant publicly announces the number of connections to other participants. Then, the one with the maximum number of connection is selected as a cluster head. Therefore, cluster head selection process is open for security threats. For instance, a malicious participant

can claim that it has the highest number of connections in its neighbourhood. Then, the malicious participant can control all the communications of the cluster. A solution for this problem is to announce the list of the connected participants with the number of connections in a secure manner, which is our final motivation for this study.

CHAPTER 2

LITERATURE SURVEY

S.Boubiche, D.E.Boubiche, A.Bilami, and H. Toral-Cruz, "**BIG DATA CHALLENGES AND DATA AGGREGATION STRATEGIES IN WIRELESS SENSOR NETWORKS**," IEEE Access, vol. 6, pp. 20558–20571, 2018.

The emergence of new data handling technologies and analytics enabled the organization of big data in processes as an innovative aspect in wireless sensor networks (WSNs). Big data paradigm, combined with WSN technology, involves new challenges that are necessary to resolve in parallel. Data aggregation is a rapidly emerging research area. It represents one of the processing challenges of big sensor networks. This paper introduces the big data paradigm, its main dimensions that represent one of the most challenging concepts, and its principle analytic tools which are more and more introduced in the WSNs technology. The paper also presents the big data challenges that must be overcome to efficiently manipulate the voluminous data, and proposes a new classification of these challenges based on the necessities and the challenges of WSNs. As the big data aggregation challenge represents the center of our interest, this paper surveys its proposed strategies in WSNs.

A. De Bonis and U. Vaccaro, "**E-ALMOST SELECTORS AND THEIR APPLICATIONS TO MULTIPLE-ACCESS COMMUNICATION**," IEEE Trans. Inf. Theory, vol. 63, no. 11, pp. 7304–7319, Nov. 2017.

Consider a group of stations connected through a multiple-access channel, with the constraint that if at a time instant exactly one station transmits a message, then the message is successfully received by any other station, whereas if two or more

stations simultaneously transmit their messages then a conflict occurs and all messages are lost. Let us assume that n is the number of stations and that an (arbitrary) subset A of them, $|A| \leq k \leq n$, is active, that is, there are at most k stations that have a message to send over the channel. In the classical conflict resolution problem, the issue is to schedule the transmissions of each station to let every active station use the channel alone (i.e., without conflict) at least once, and this requirement must be satisfied whatever might be the set of active stations A. The parameter to optimize is, usually, the worst case number of transmissions that any station has to attempt before all message transmissions are successful. In this paper, we study the following question: is it possible to obtain a significant improvement on the protocols that solve the classical conflict resolution problem if we allow the protocols to fail over a "small" fraction of all possible subsets of active stations? In other words, is it possible to significantly reduce the number of transmissions that must be attempted if the set of active stations is chosen uniformly at random and the conflict resolution algorithm is only required to work correctly with "high" probability? In this paper, we will show that this is indeed the case. Our main technical tool is a generalization of selectors, a recently introduced combinatorial structure that has found applications in several areas. As it turned out for selectors, we believe that our new combinatorial structures are likely to be useful also outside the present context.

C. Aksoylar, G. K. Atia, and V. Saligrama, "**SPARSE SIGNAL PROCESSING WITH LINEAR AND NONLINEAR OBSERVATIONS: A UNIFIED SHANNON-THEORETIC APPROACH**," IEEE Trans. Inf. Theory, vol. 63, no. 2, pp. 749–776, Feb. 2017

We derive fundamental sample complexity bounds for recovering sparse and structured signals for linear and nonlinear observation models including sparse

regression, group testing, multivariate regression and problems with missing features. In general, sparse signal processing problems can be characterized in terms of the following Markovian property. We are given a set of N variables X1, X2, . . . XN, and there is an unknown subset of variables S ⊂ {1. . . N} that are relevant for predicting outcomes Y. More specifically, when Y is conditioned on {Xn}n∈S it is conditionally independent of the other variables, {Xn}n6∈S. Our goal is to identify the set S from samples of the variables X and the associated outcomes Y. We characterize this problem as a version of the noisy channel coding problem. Using asymptotic information theoretic analyses, we establish mutual information formulas that provide sufficient and necessary conditions on the number of samples required to successfully recover the salient variables. These mutual information expressions unify conditions for both linear and nonlinear observations. We then compute sample complexity bounds for the aforementioned models, based on the mutual information expressions in order to demonstrate the applicability and flexibility of our results in general sparse signal processing models.

C.-J. Liu, P. Huang, and L. Xiao, "**TAS-MAC: A TRAFFIC-ADAPTIVE SYNCHRONOUS MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS**," IEEE, ACM Trans. Sensor Network., vol. 12, no. 1, p. 1, 2016.

Duty cycling improves energy efficiency but limits throughput and introduces significant end-to-end delay in wireless sensor networks. In this article, we present a traffic-adaptive synchronous MAC protocol (TASMAC), which is a high-throughput, low-delay MAC protocol tailored for low power consumption. It achieves high throughput by adapting time division multiple access (TDMA) to a novel traffic-adaptive allocation mechanism that assigns time slots only to nodes located on active routes. TAS-MAC reduces the end-to-end delay by notifying all

nodes on active routes of incoming traffic in advance. These nodes will claim time slots for data transmission and forward a packet through multiple hops in a cycle. The desirable traffic-adaptive feature is achieved by decomposing traffic notification and data-transmission scheduling into two phases, specializing their duties and improving their efficiency, respectively. Simulation results and experiments on TelosB motes demonstrate that the two-phase design significantly improves the throughput of current synchronous MAC protocols and achieves the similar low delay of slot-stealing-assisted TDMA with much lower power consumption

S. Wu, S. Wei, Y. Wang, R. Vaidyanathan, and J. Yuan, "**PARTITION INFORMATION AND ITS TRANSMISSION OVER BOOLEAN MULTI-ACCESS CHANNELS**," IEEE Trans. Inf. Theory, vol. 61, no. 2, pp. 1010–1027, Feb. 2015.

In this paper, we propose a novel reservation system to study partition information and its transmission over a noise free Boolean multiaccess channel. The objective of transmission is not to restore the message, but to partition active users into distinct groups so that they can, subsequently, transmit their messages without collision. We first calculate (by mutual information) the amount of information needed for the partitioning without channel effects, and then propose two different coding schemes to obtain achievable transmission rates over the channel. The first one is the brute force method, where the codebook design is based on centralized source coding; the second method uses random coding, where the codebook is generated randomly and optimal Bayesian decoding is employed to reconstruct the partition. Both methods shed light on the internal structure of the partition problem. A novel formulation is proposed for the random coding scheme, in which a sequence of channel operations and interactions induces a hypergraph. The

formulation intuitively describes the transmitted information in terms of a strong colouring of this hypergraph. An extended Fibonacci structure is constructed for the simple, but nontrivial, case with two active users. A comparison between these methods and group testing is conducted to demonstrate the potential of our approaches.

M. L. Malloy and R. D. Nowak, "**NEAR-OPTIMAL ADAPTIVE COMPRESSED SENSING**," IEEE Trans. Inf. Theory, vol. 60, no. 7, pp. 4001–4012, Jul. 2014

This paper proposes a simple adaptive sensing and group testing algorithm for sparse signal recovery. The algorithm, termed compressive adaptive sense and search (CASS), is shown to be near-optimal in that it succeeds at the lowest possible signal-to-noise-ratio (SNR) levels, improving on previous work in adaptive compressed sensing. Like traditional compressed sensing based on random nonadaptive design matrices, the CASS algorithm requires only k log n measurements to recover a k-sparse signal of dimension n. However, CASS succeeds at SNR levels that are a factor log n less than required by standard compressed sensing. From the point of view of constructing and implementing the sensing operation as well as computing the reconstruction, the proposed algorithm is substantially less computationally intensive than standard compressed sensing. The CASS is also demonstrated to perform considerably better in practice through simulation. To the best of our knowledge, this is the first demonstration of an adaptive compressed sensing algorithm with near-optimal theoretical guarantees and excellent practical performance. This paper also shows that methods like compressed sensing, group testing, and pooling have an advantage beyond simply reducing the number of measurements or tests— adaptive versions of such methods can also improve detection and estimation performance when compared

with nonadaptive direct (uncompressed) sensing.

V. Y. F. Tan and G. K. Atia, "**STRONG IMPOSSIBILITY RESULTS FOR SPARSE SIGNAL PROCESSING**," IEEE Signal Process. Let. vol. 21, no. 3, pp. 260–264, Mar. 2014

This letter derives strong impossibility results for several sparse signal processing problems. It is shown that regardless of the allowed error probability in identifying the salient support set (as long as this probability is below one), the required number of measurements is almost the same as that required for the error probability to be arbitrarily small. Our proof technique involves the use of the blowing-up lemma and can be applied to diverse problems from noisy group testing to graphical model selection as long as the observations are discrete.

C. Lam Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "**NON-ADAPTIVE GROUP TESTING: EXPLICIT BOUNDS AND NOVEL ALGORITHMS**," IEEE Trans. Inf. Theory, vol. 60, no. 5, pp. 3019–3035, May 2014.

We consider some computationally efficient and provably correct algorithms with near-optimal sample complexity for the problem of noisy nonadaptive group testing. Group testing involves grouping arbitrary subsets of items into pools. Each pool is then tested to identify the defective items, which are usually assumed to be sparse. We consider nonadaptive randomly pooling measurements, where pools are selected randomly and independently of the test outcomes. We also consider a model where noisy measurements allow for both some false negative and some false positive test outcomes (and also allow for asymmetric noise, and activation noise). We consider three classes of algorithms for the group testing problem (we call them specifically the coupon collector algorithm, the column matching algorithms, and the LP decoding algorithms—the last two classes of algorithms (versions of some of which had been considered before in the literature) were

inspired by corresponding algorithms in the compressive sensing literature. The second and third of these algorithms have several flavors, dealing separately with the noiseless and noisy measurement scenarios. Our contribution is novel analysis to derive explicit sample-complexity bounds—with all constants expressly computed—for these algorithms as a function of the desired error probability, the noise parameters, the number of items, and the size of the defective set (or an upper bound on it). We also compare the bounds to information-theoretic lower bounds for sample complexity based on Fano's inequality and show that the upper and lower bounds are equal up to an explicitly computable universal constant factor (independent of problem parameters).

P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "**THE EVOLUTION OF MAC PROTOCOLS IN WIRELESS SENSOR NETWORKS: A SURVEY**," IEEE Communication Surveys Tuts., vol. 15, no. 1, pp. 101–120, 1st Quart., 2013.

Wireless Sensor Networks (WSNs) have become a leading solution in many important applications such as intrusion detection, target tracking, industrial automation, smart building and so on. Typically, a WSN consists of a large number of small, low-cost sensor nodes that are distributed in the target area for collecting data of interest. For a WSN to provide high throughput in an energy-efficient way, designing an efficient Medium Access Control (MAC) protocol is of paramount importance because the MAC layer coordinates nodes' access to the shared wireless medium. To show the evolution of WSN MAC protocols, this article surveys the latest progresses in WSN MAC protocol designs over the period 2002-2011. In the early development stages, designers were mostly concerned with energy efficiency because sensor nodes are usually limited in power supply. Recently, new protocols are being developed to provide multitask support and efficient delivery of bursty traffic. Therefore, research attention has turned back to

throughput and delay. This article details the evolution of WSN MAC protocols in four categories: asynchronous, synchronous, frame-slotted, and multichannel. These designs are evaluated in terms of energy efficiency, data delivery performance, and overhead needed to maintain a protocol's mechanisms. With extensive analysis of the protocols many future directions are stated at the end of this survey. The performance of different classes of protocols could be substantially improved in future designs by taking into consideration the recent advances in technologies and application demands.

G. K. Atia and V. Saligrama, "**BOOLEAN COMPRESSED SENSING AND NOISY GROUP TESTING**," IEEE Trans. Inf. Theory, vol. 58, no. 3, pp. 1880–1901, Mar. 2012

The fundamental task of group testing is to recover a small distinguished subset of items from a large population while efficiently reducing the total number of tests (measurements). The key contribution of this paper is in adopting a new information

theoretic perspective on group testing problems. We formulate the group testing problem as a channel coding/decoding problem and derive a single-letter characterization for the total number of tests used to identify the defective set. Although the focus of this paper is primarily on group testing, our main result is generally applicable to other compressive sensing models. Furthermore, our bounds allow us to verify existing known bounds for noiseless group testing including the deterministic noise-free case and approximate reconstruction with bounded distortion. Our proof of achievability is based on random coding and the analysis of a maximum likelihood detector, and our information theoretic lower bound is based on Fano's inequal.

CHAPTER 3

SYSTEM DESIGN

3.1 EXISTING SYSTEM

Overview of the Literature Survey

Numerous Medium Access Control (MAC) protocols for WSN have been suggested over the years, designed to cope with various setups and objectives. In particular, traffic in a WSN can be quite dynamic, depending on the events being sensed, the sensing application and the protocols being used.

Therefore, such protocols should perform well under a wide range of traffic loads, a variety of network topologies and various objectives such as latency, reliability, energy consumption, security, etc. Accordingly, many of these MAC protocols were designed and examined to be robust under diverse setups. The approach of designing a MAC protocol regardless of the wireless Phy-protocol being used, the routing protocol employed, and even the application expected to utilize it, is consistent with the network-layering conceptual model. However, even though these protocols perform well under a large variety of setups, this universality comes at a price.

For example, a MAC protocol allowing any subset of the sensors, regardless of its size, to transmit simultaneously will suffer from many collisions and high overhead if indeed a large subset attempts transmission concurrently. Another example is a keep alive frame transmitted over Wi-Fi, that conveys only the sender ID and a single bit of relevant information (the sensor is alive), yet requires a large number of bits to be transferred due to headers, physical encapsulation, etc.

We divide the discussion on existing system into two parts. In the first, we provide a brief overview of several multipurpose WSN MAC protocols.

Then, since our protocol is inspired by the classical Group Testing (GT) approach, in the second part we give a brief overview of related GT results.

WSN MAC PROTOCOLS

Since on the one hand, one of the foremost objectives of WSN is energy conservation, and on the other sensor nodes are expected to report only sporadically (and many of the reports can tolerate a short delay), most of the MAC protocols which were designed for WSN over the past decade and a half rely on a duty cycling technique in which each sensor node turns its radio on only periodically, alternating between active and sleeping modes. Such protocols took different approaches to address the rendezvous challenge in which a sender and a receiver should be awake at the same time in order to exchange information. In the synchronous approach, nodes' active and sleeping periods are aligned, i.e., all sensor nodes are active at the same time intervals and are required to contend for transmission opportunities during these intervals.

The asynchronous approach allows sensor nodes to choose individual wakeup times, maintaining unsynchronized duty-cycles, and employing various strategies to detect transmissions in the network and enable rendezvous between senders and receivers. However, all these protocols are designed to support various types of traffic patterns, diverse topologies (e.g., single and multi-hop topologies) and most importantly, to support any kind of information exchange between the sensors. Accordingly, they have adopted the traditional approach in which the proposed channel access mechanism is independent from the message payload exchange between the sensor nodes, at the price of data encapsulation and signalling overhead. In this work, since the information each sensor needs to convey is limited to one out of a number of known messages, we take a cross-layer design in which the coding and the channel access algorithm are

intertwined. Indeed recently, data aggregation was used to compress the data and reduce its redundancy in order to save network energy before transmission to the sink. In a sense, in the protocol we give herein, using the coding we suggest, the channel inherently compresses the data efficiently (based on a modified GT approach) by the addition in the air of the K messages transmitted simultaneously from the sensors to the sink.

CHAPTER 4

PROPOSED SYSTEM

4.1 INTRODUCTION

Our project is to detect the malicious node in the network with less energy consumption, less time delay and more throughput and enhance the performance of WSN. In our existing paper, we have used Signature based intrusion detection system which use different techniques such as Principle component analysis for analysing the node, Singular value decomposition for data splitting, Gaussian naive bayes for detecting the malicious node, Stochastic gradient descent which classifies the routing nodes and malicious node for further data transmission. So for each step it consumes more energy and time for detecting the malicious node.

In our project, we have used SET PROTOCOL and RSA algorithm which uses only two techniques for detecting the malicious node and all processes. So the detection time and energy is reduced compared to the existing protocol. In RSA algorithm we use key management system which generates public key, private key (secret key) and master key (mk). We propose the network model that contains some clusters. Each cluster has its coordinator namely Cluster Head(CH). The clusters are interconnected via CHs.

Our new key management scheme namely "Secure and Efficient Transmission (SET)" is a streamlined method for managing group keys, leveraging multiple tree-based multicast routing schemes to enhance reliability by utilizing diverse communication paths. So, in our protocol, two multicast trees are used for each subgroup. For example, in a cluster, the connection of multicast tree is maintained as its CH that computes and distributes the intermediate keying materials to all members in this cluster through the active

tree links. Also, the CH is responsible for maintaining the connection of the multicast subgroup.

- A new MAC protocol, for data collection from dense wireless sensor networks, in which the sensors are expected to transmit only sporadically, and a predefined amount of information
- In the suggested protocol, the sink can collect up to K reports simultaneously without any management or scheduling.
- We present a downlink version of the protocol as well, that is suitable for data dissemination from a sink to a set of K out of N sensors simultaneously, without any management overhead, a predefined schedule or even a message notifying the relevant set of sensors.
- To support the protocols, we provide a codebook construction with a very simple encoding and decoding procedure, such that, not only the code is efficient but also the transmitted codewords are self-contained and do not require headers, trailers or sender identity. Moreover, we suggest effective decoding algorithms based on Column Matching.
- We extend the single hop setup considered throughout the paper a to multi-hop setup in which a message needs to traverse multiple hops (pass through multiple relays) before reaching its designated sink.
- We propose a secure and efficient transmission for cluster-based group key protocol for WSNs, namely SGKP (Secure Group Key Protocol) by improving the security of protocol in and by adding a new dynamic group operation called the cluster merge operation.
- We propose a novel secure cluster head selection mechanism for SGKP-WSN.
- SGKP provides security against the known-key attacks defined and better performance in terms of reducing the communications cost and computational cost of computing and updating group key.

- Also provides efficient and secure group key computation solution by eliminating the security and performance issues in two-level group key agreement protocols for WSNs.
- An example application scenario for SGKP on a disaster area communication with simulation is presented.

4.2 PROPOSED BLOCK DIAGRAM

In WSN monitoring system, the sensor node is fixed at the place of interest for long duration to gather the surrounding information. In some cases, the duration could be up to a year without any maintenance, and thus require efficient power consumption. Figure shows the block diagram of sensor node consists of few cores including sensory, computational and communication.

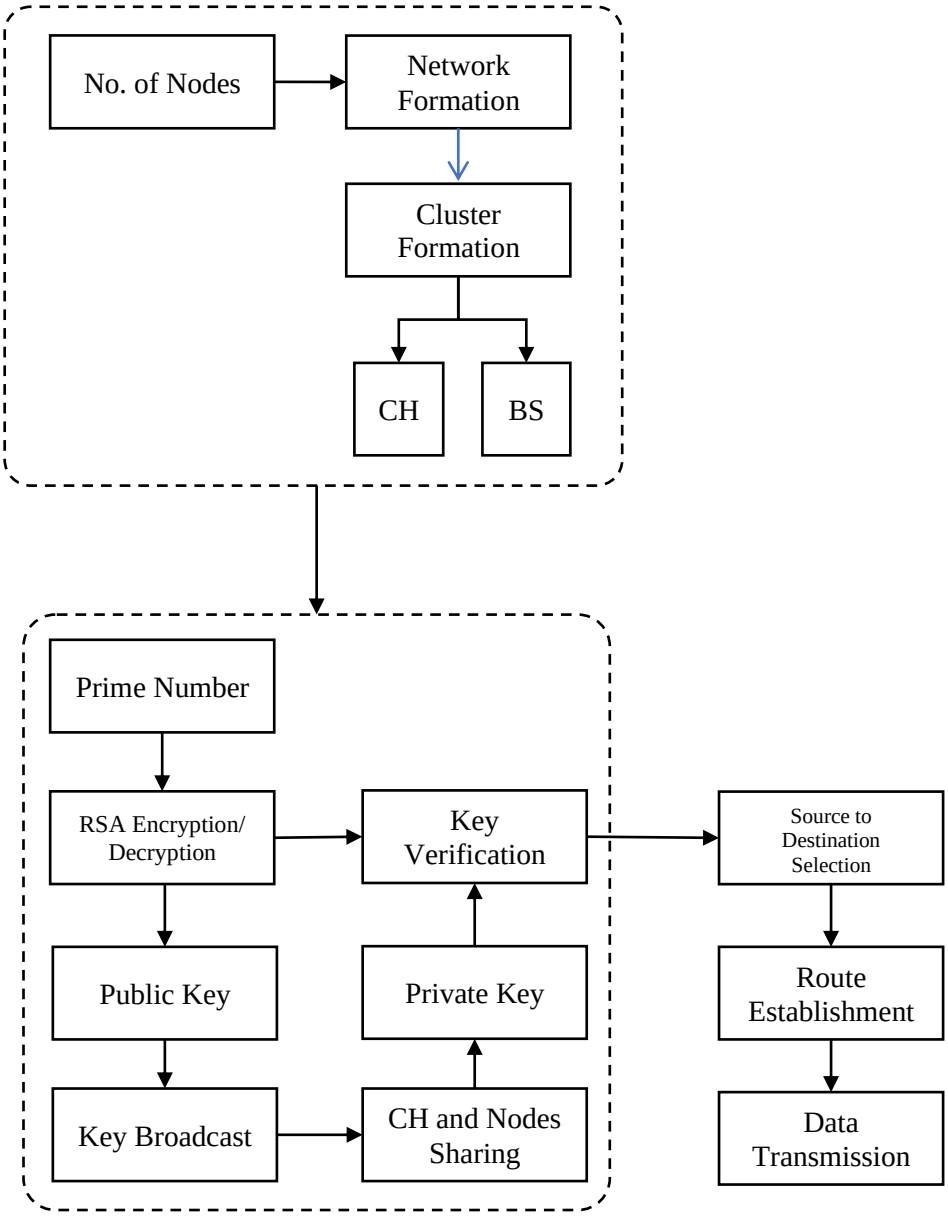


Fig.4.1 Proposed Block diagram

We proposed a new approach which aims to address the scalability problem while taking into consideration the dynamic aspect of the group members and dynamicity of nodes in WSN. There are two trees on the network to avoid the robustness problem as well. Our approach is based on clustering manner. Each cluster is initiated by Cluster Heads.

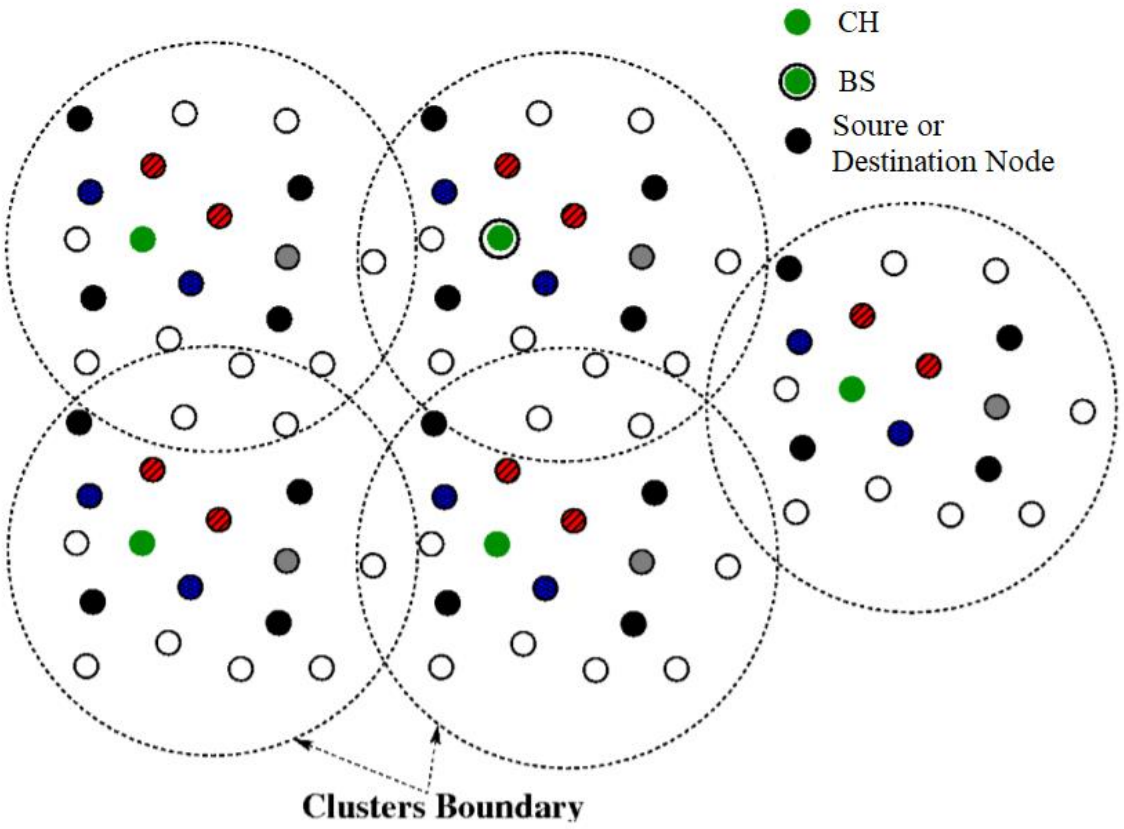


Fig.4.2 Cluster based Data Aggregation

CH has two keys, one for its cluster subgroup and another one for the interconnection among the clusters via CHs.

There are many multicast routing protocols have been proposed, these protocols are classified as shown before. We proposed another one in the category of multicast topology, tree-based and shared tree with double trees, namely Blue tree and Red tree. All clusters then work in parallel to construct two trees. Logically, a group member views the two trees as identical trees. The group members have to be in both multicast trees inside the Cluster. In a cluster, CH starts to initialize the process for a cluster multicast subgroup by broadcasting a join advertises message across the entire cluster. This cluster is bounded and having a fixed diameter. Each node is associated with three colours (blue, red, and grey). A node will choose its colour (grey) when its total

number of neighbours is less than a predefined threshold value (depending on average node degree, for instance, half of its degree). Other nodes randomly choose blue or red as their colour with probability equal to 0.5. For the first received message, a grey node stores the upstream node ID and rebroadcasts the message except the node that the message is coming from. For a non-grey node, it stores the upstream node ID and rebroadcasts the message only if the upstream node is the same colour, a sender/receiver, or a grey node.

INTERCONNECTION AMONG THE CLUSTERS

The interconnection among the clusters is via the MCH (Master Cluster Heads) starts to initialize the process for a CH's multicast subgroup by broadcasting a join advertises message across the entire WSN. Nodes maintain their assigned colours (blue, red, grey), with Cluster Heads (CHs) acting as both sources and receivers. These CHs form a virtual cluster for data transmission. So, we can apply the same scenario that is used before in the cluster, to get blue and red multicast trees among all CHs in WSN.

4.3 GROUP KEY ESTABLISHMENT PROTOCOL

The idea of subgroup key agreement protocol is that all subgroup members maintain a logic key's tree in local storage space. This key's tree is used to deduce the final common subgroup key. Before introducing the security and performance properties of group key agreement protocols and the detailed definition of SGKP, we give the basic definitions related to cluster node, cluster head, public parameters and cluster concepts for the use of group key agreement protocols in WSNs.

An SET scheme is implemented for SGKPs consists of the following operations, specifically, setup at the BS, key extraction and signature signing at

the data sending nodes, and verification at the data receiving nodes:

- **Setup**. The BS (as a trust authority) generates a master key (mk) and public parameters for the private key generator (PKG), and gives them to all sensor nodes.
- **Extraction**. Given an ID string, a sensor node generates a private key associated with the ID using master key.
- **Signature signing**. Given a message (M), time stamp (t) and a signing key (sk), the sending node generates a signature (SIG).
- **Verification**. Given the ID, Message, and Signature, the receiving node outputs "accept" if SIG is valid, and outputs "reject" otherwise.

The proposed SET-SGKP has a protocol initialization prior to the network deployment and we introduce the protocol initialization, describing the key management of the protocol by using the SET protocol, and the protocol operations afterwards.

4.4 PROTOCOL INITIALIZATION

In SET-SGKP, time is divided into successive time intervals as other denote time stamps by 'Ts' for BS-to-node communication and by 'tj' for leaf-to-CH communication. Note that key pre-distribution is an efficient method to improve communication security.

In this project, we adopt ID 'tk' as user's public key under SET protocol, and propose a secure data transmission protocol by using SET specifically for SGKPs. The corresponding private pairing parameters are preloaded in the sensor nodes during the protocol initialization.

In this way, when a sensor node wants to authenticate itself to another node, it does not have to obtain its private key at the beginning of a new round. Upon node revocation, the BS broadcasts the compromised node IDs to all sensor nodes, each node then stores the revoked IDs within the current round.

We adopt the additively RSA (Rivest–Shamir–Adleman) encryption scheme to encrypt the random number of node data, in which a specific operation performed on the plaintext is equivalent to the operation performed on the composite number.

Using this protocol that allows efficient aggregation of encrypted data at the CHs and the BS, which also guarantees data confidentiality. In the protocol initialization, the BS performs the following operations of key pre-distribution to all the sensor nodes:

- Generate an encryption key k for the RSA encryption scheme to encrypt data messages, where k = [m-1], m is a large integer.
- Generate the pairing parameters and generate stochastically. Choose two cryptographic prime functions: H, for the point mapping hash function which maps strings to elements, and h, for mapping arbitrary inputs to fixed-length outputs.
- Pick a random integer as the master key (mk), Private key (setk) and set P as network public key. Preload each sensor node with the system parameters.

PSEUDO CODE: RSA

Key generation

1. Consider 6 large prime numbers p, q, r, s, t, and u.
2. Compute n=p*q*r*s*t*u.
3. Compute $\varphi(n)$= (p-1)*(q-1)*(r-1)*(s-1)*(t-1)*(u-1), where φ is Euler's totient function. This value is kept private.
4. Choose an integer *e* such that *1 < e <φ (n) and gcd (e, φ (n)) = 1*; i.e., *e* and *φ (n)* are co-prime.
5. Find d, such that d*e mod $\varphi(n)$=1.

   Publish e and n as the public key. (Or) public key= {e, n}

   Keep d and m as the secret key. (Or) private key= {d, n}

Encryption

C= m^e(mod n)

Decryption

M= c^d(mod n)

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 / 8.1
- Virtualization : VMware 9.0 & Virtualized Red Hat Linux
- Software Tool : Network Simulator 2 (NS2)
- Software Package : ns-allinone-2.34
- Languages : Tcl (Tool Command Language), OTcl (Object Tcl)

5.2 HARDWARE REQUIREMENTS

- Processor : Any Intel or AMD x86/x64 processor
- RAM : 1024MB (At least 2048 MB recommended)
- Disk Space : 10–12 GB for a Virtualized Red Hat installation
- Graphics : No specific graphics card is required.

5.3 SOFTWARE DESCRIPTION:

SIMULATIONS:

Most of the commercial simulators are GUI driven, while some network simulators require input scripts or commands (network parameters). The network parameters describe the state of the network (node placement, existing links) and

the events (data transmission, link failures, etc).

An important output of simulations is the trace files. Trace files can document every event that occurred in the simulation and are used for analysis. Certain simulators have added functionality of capturing the type of data directly from a functioning production environment, at various times of the day, week, or month, in order to reflect average, worst-case, and best-case conditions. Network simulators can also provide others tools to facilitate visual analysis of trends and potential trouble spots.

Network simulators, as the name suggests are used by researchers, developers and QA to design various kinds of networks, simulate and then analyze the effect of various kinds of networks, simulate and then analyze the effect of various parameters on the network performance.

A typical network simulator encompasses a wide range of networking technologies and helps the users to build complex networks from basic building blocks like variety of nodes and links. With the help of simulators one can design hierarchical networks using various types of nodes like computers, hubs, bridges, routers, optical cross-connects, multicast router, mobile units, MSAUs etc.

5.4 SIMULATION TECHNIQUES

Most network simulators use discrete event simulation, in which a list of pending “events” is stored, and those events are processed in order, with some events triggering future events such as the event of the arrival of a packet at one node triggering the event of the arrival of that packet at a downstream node. Some network simulation problems, notably those relying on queuing theory, are well suited to Markov chain simulation in which no list of future events is maintained and the simulation consists of transiting between different system “states” in a

memory less fashion . Markov chain simulation is typically faster but less accurate and flexible then detailed discrete event simulation. Some simulation is cyclic based simulations and these are faster as compared to event based simulations.

5.5 NS-2 (SIMULATOR)

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

In 1996-97, ns version 2 (ns-2) was initiated based on a refactoring by Steve McCann. Use of Tcl was replaced by MIT's Object Tcl (OTcl), an object-oriented dialect Tcl. The core of ns-2 is also written in C++, but the C++ simulation objects are linked to shadow objects in OTcl and variables can be linked between both language real ms. Simulation scripts are written in the OTcl language, an extension of the Tcl scripting language.

Presently, ns-2 consists of over 300,000 lines of source code, and there is probably a comparable amount of contributed code that is not integrated directly into the main distribution. It runs on GNU/Linux, FreeBSD, Solaris, Mac OS X and Windows versions that support Cygwin. It is licensed for use under version 2 of the GNU General Public License.

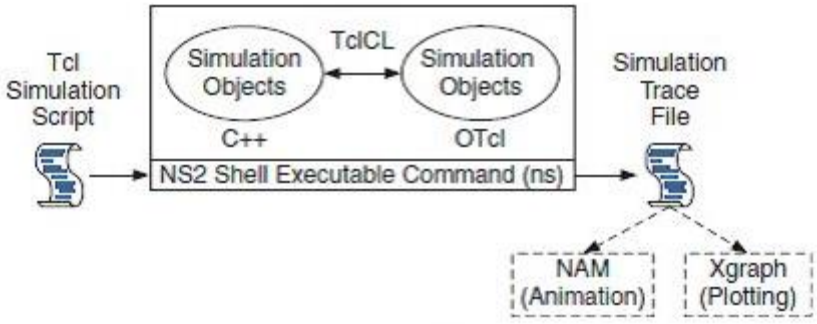
Features of NS2

- It is a discrete event simulator for networking research.
- It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, HTTP and DSR.
- It simulates wired and wireless network.
- It is primarily UNIX based.
- Uses TCL as its scripting language.

- OTcl: Object oriented support.
- Tcl: C++ and OTcl linkage.
- Discrete event scheduler.

5.6 BASIC ARCHITECTURE

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend)



Basic architecture of NS.

Fig. 5.1 Basic Architecture of NS-2

Simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using Tcl.

Language (TCL and C++)

NS2 uses OTcl to create and configure a network, and uses C++ to run simulation. All C++ codes need to be compiled and linked to create an executable file.

Uses of OTcl

- For configuration, setup, or one time simulation.
- To run simulation with existing NS2 modules.

This option is preferable for most beginners, since it does not involve complicated internal mechanism of NS2. Unfortunately, existing NS2 modules are fairly limited. This option is perhaps not sufficient for most researchers.

Uses of C++

- When you are dealing with a packet, or when you need to modify existing NS2 modules.

This option discourages most of the beginners from using NS2. This book particularly aims at helping the readers understand the structure of NS2 and feel more comfortable in modifying NS2 modules.

Tools for generating TCL Script for NS2

NS2a very common and widely used tool to simulate small and large area networks. Tcl scripts are widely used inNS-2 simulation tool. Tcl scripts are used to set up a wired or wireless communication network, and then run these scripts via the NS-2 for getting the simulation results. Several tools are available to design networks and generate TCL scripts some of them are discussed below

I. NS2 scenario Generator (NSG):

It’s Java based tool that can run on any platform and can generate TCL scripts for wired and Wireless scenarios for NS2.Main features of NSG are:

1. Creating Wired and wireless nodes by drag and drop.
2. Creating Simplex and Duplex links for wired network.
3. Creating Grid, Random and Chain topologies.
4. Creating TCP and UDP agents.
5. Also supports Tahoe, TCP Reno, TCP New-Reno and TCP Vegas.
6. Supports Ad Hoc routing protocols such as DSDV, AODV, DSR and TORA.
7. Supports FTP and CBR applications.
8. Supports node mobility.
9. Setting the packet size, start time of simulation, end time of simulation, transmission range and interference range in case of wireless networks, etc.
10. Setting other network parameters such as bandwidth, etc for wireless scenarios.

Trace Files Generated in NS2

NS2 currently supports a number of different types of trace files. In addition to its own format, NS2 also has the Nam trace format, which contains the necessary information from the simulation to drive the Nam visualize. Both of these trace formats are very specific when it comes to giving details about the events that occur during an NS2 simulation. Traces and monitors represent the only support for data collection in ns-2. Traces record events related to the generation, enqueueing, forwarding, and dropping of packets. Each event corresponds to a line of ASCII characters, which contains information on the event type and the

information stored into the packet. NS-2 provides three kinds of formats for wired networks: Tracing, Monitoring and NAM trace file.

Tracing: Trace file format is given below:

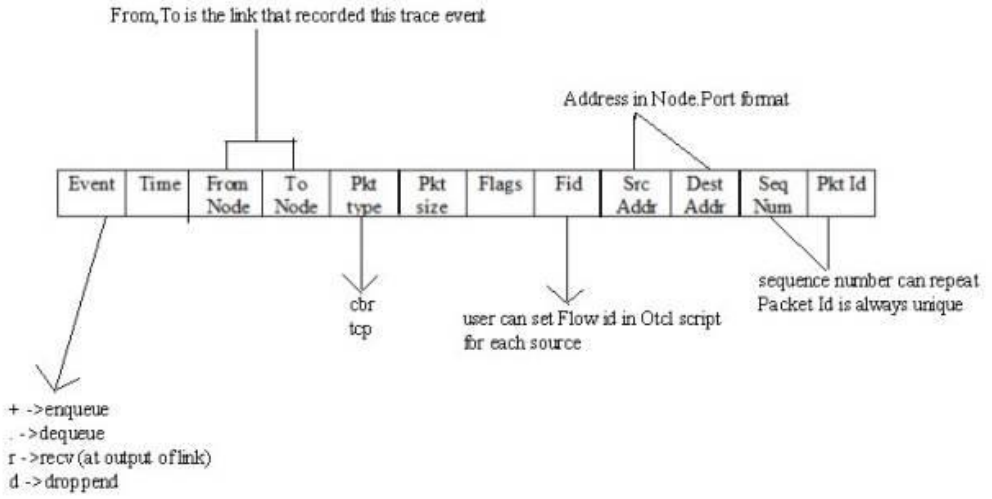


Fig.5.2 Trace File Format

1. Operation performed in the simulation.
2. Simulation time of event occurrence.
3. Node 1 of what is being traced.
4. Node 2 of what is being traced.
5. Packet type.
6. Packet size.
7. Flags.
8. IP flow identifier.
9. Packet source node address.
10. Packet destination node address.
11. Sequence number.
12. Unique packet identifier.

Example of Trace file:

```
+ 4.315145 0 1 ack 40 ------- 0 3.0 6.0 0 1
- 4.315145 0 1 ack 40 ------- 0 3.0 6.0 0 1
r 4.325305 0 1 ack 40 ------- 0 3.0 6.0 0 1
+ 4.325305 1 6 ack 40 ------- 0 3.0 6.0 0 1
- 4.325305 1 6 ack 40 ------- 0 3.0 6.0 0 1
r 4.335465 1 6 ack 40 ------- 0 3.0 6.0 0 1
+ 4.335465 6 1 tcp 1040 ------- 0 6.0 3.0 1 2
- 4.335465 6 1 tcp 1040 ------- 0 6.0 3.0 1 2
r 4.349625 6 1 tcp 1040 ------- 0 6.0 3.0 1 2
+ 4.349625 1 0 tcp 1040 ------- 0 6.0 3.0 1 2
- 4.349625 1 0 tcp 1040 ------- 0 6.0 3.0 1 2
r 4.363785 1 0 tcp 1040 ------- 0 6.0 3.0 1 2
+ 4.363785 0 4 tcp 1040 ------- 0 6.0 3.0 1 2
- 4.363785 0 4 tcp 1040 ------- 0 6.0 3.0 1 2
r 4.377945 0 4 tcp 1040 ------- 0 6.0 3.0 1 2
+ 4.377945 4 3 tcp 1040 ------- 0 6.0 3.0 1 2
- 4.377945 4 3 tcp 1040 ------- 0 6.0 3.0 1 2
r 4.392105 4 3 tcp 1040 ------- 0 6.0 3.0 1 2
+ 4.392105 3 4 ack 40 ------- 0 3.0 6.0 1 3
- 4.392105 3 4 ack 40 ------- 0 3.0 6.0 1 3
r 4.402265 3 4 ack 40 ------- 0 3.0 6.0 1 3
```

Fig. 5.3 Examples of Trace Files

II. Monitoring

Queue monitoring refers to the capability of tracking the dynamics of packets at a queue (or other object).

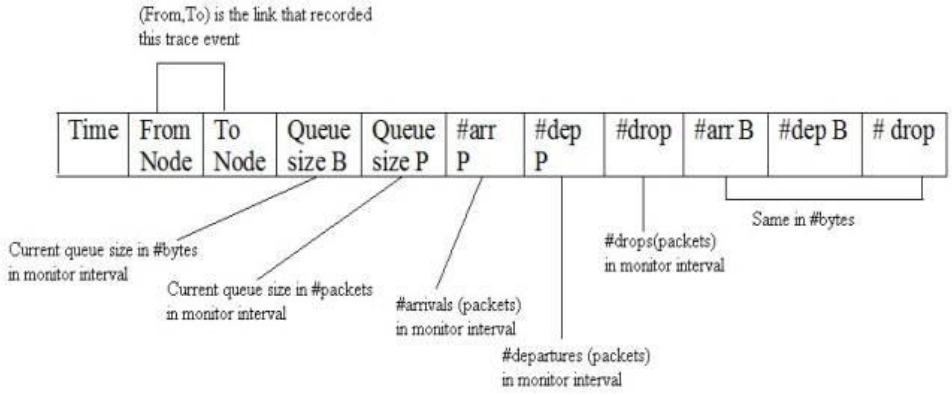


Fig. 5.4 Flow Trace Format

A queue monitor tracks packet arrival/departure/drop statistics, and may optionally compute averages of these values. Monitoring was useful tools to find detail information about queue.

NAM trace files which are used by NAM for visualization of ns simulations

The NAM trace file should contain topology information like nodes, links, queues, node connectivity etc as well as packet trace information. A NAM trace file has a basic format to it. Each line is a NAM event. The first character on the line defines the type of event and is followed by several flags to set options on that event. There are 2 sections in that file, static initial configuration events and animation events. All events with -t * in them are configuration events and should be at the beginning of the file.

```
n -t * -a 4 -s 4 -S UP -v circle -c black -i black
n -t * -a 0 -s 0 -S UP -v circle -c black -i black
n -t * -a 5 -s 5 -S UP -v circle -c black -i black
n -t * -a 1 -s 1 -S UP -v circle -c black -i black
n -t * -a 6 -s 6 -S UP -v circle -c black -i black
n -t * -a 2 -s 2 -S UP -v circle -c black -i black
n -t * -a 3 -s 3 -S UP -v circle -c black -i black
l -t * -s 2 -d 4 -S UP -r 2000000 -D 0.01 -c black
l -t * -s 3 -d 4 -S UP -r 2000000 -D 0.01 -c black
l -t * -s 4 -d 0 -S UP -r 2000000 -D 0.01 -c black
l -t * -s 0 -d 1 -S UP -r 2000000 -D 0.01 -c black
l -t * -s 6 -d 1 -S UP -r 2000000 -D 0.01 -c black
l -t * -s 1 -d 5 -S UP -r 2000000 -D 0.01 -c black
+ -t 4.25418515323951 -s 6 -d 1 -p tcp -e 40 -c 0 -i 0 -a 0 -x
{6.0 3.0 0 ------- null}
- -t 4.25418515323951 -s 6 -d 1 -p tcp -e 40 -c 0 -i 0 -a 0 -x
{6.0 3.0 0 ------- null}
h -t 4.25418515323951 -s 6 -d 1 -p tcp -e 40 -c 0 -i 0 -a 0 -x
{6.0 3.0 -1 ------- null}
r -t 4.26434515323951 -s 6 -d 1 -p tcp -e 40 -c 0 -i 0 -a 0 -x
{6.0 3.0 0 ------- null}
+ -t 4.26434515323951 -s 1 -d 0 -p tcp -e 40 -c 0 -i 0 -a 0 -x
{6.0 3.0 0 ------- null}
- -t 4.26434515323951 -s 1 -d 0 -p tcp -e 40 -c 0 -i 0 -a 0 -x
{6.0 3.0 0 ------- null}
```

Fig. 5.5 NAM Trace Files

Example of NAM file is:

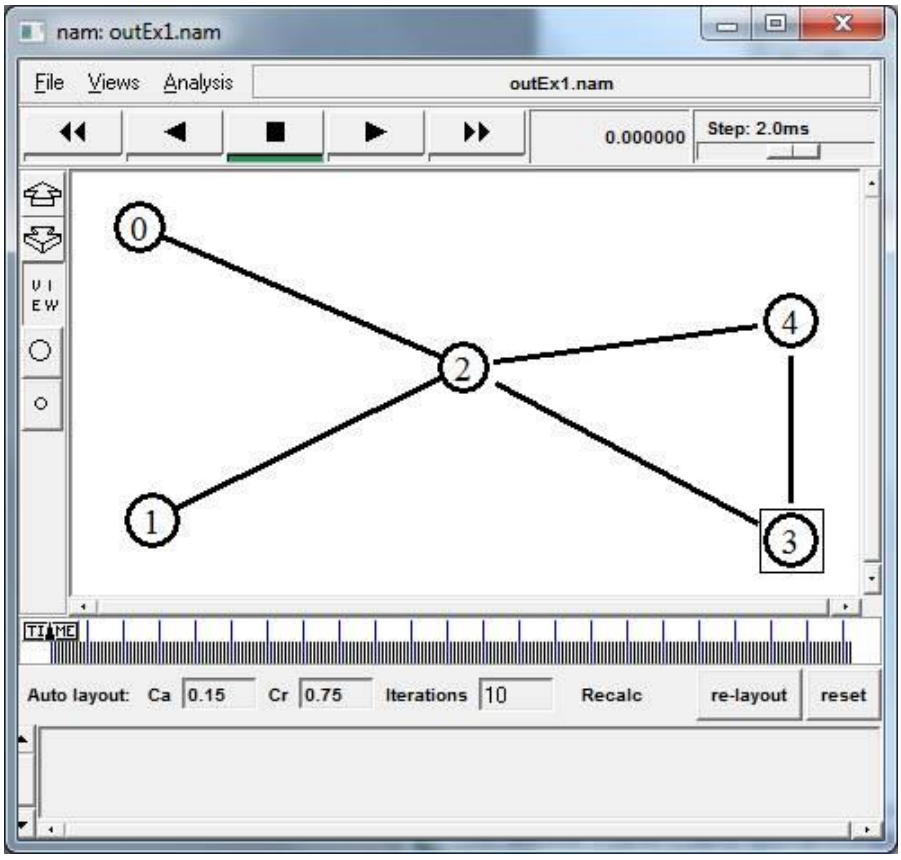


Fig. 5.6 NAM Window

Ns or the Network simulator (also popularly called ns-2) is a discrete event network simulator. It is popular in academia for its extensibility (due to its open source model) and plentiful online documentation. Ns is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc networking research. Ns supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. It can be also used as limited –functionality network emulator. Ns is licensed for use under version 2 of the GNU General Public License.

ABOUT - TCL

Tool Command Language (TCL) is an interpreted script language developed by Dr.John Ouster out at the University of California, Berkeley, and now developed and maintained by Scrip tics. Tcl is comparable to: Netscape JavaScript Microsoft's Visual Basic. The UNIX-derived Practical Extraction and Reporting Language IBM's Restructured Extended Executor In general, script languages are easier and faster to code in than the more structured, complied languages such as C and C++. Script languages are sometimes considered good "glue "languages for tying several compiled programs together. Or, as stand-alone programs, they can allow you to create simple but powerful effects on their own. Tcl Blend is a version of Tcl that can access certain Java languages facilities. Tcl has a companion program, Tool kit (Tk), to help create a Graphical User Interface with Tcl.

OTCL

OTCL is an object oriented extension of Tcl and created by David Wetherill. It is used in network simulator (NS-2) and usually run under UNIX environment.

GEDIT

GEDIT is a UTF-8 compatible text editor for the GNOME computer desktop environment. Designed as a general purpose text editor, gedit emphasizes simplicity and ease of use. It includes tools for editing source code and structured text such as mark-up languages. It is designed to have a clean, simple graphical user interface according to the philosophy of the GNOME project, and it is the default text editor for GNOME.

Gedit includes syntax highlighting for various program code and text mark-up formats. Gedit also has GUI tabs for editing multiple files. Tabs can be moved

between various windows by the user. It can edit remote files using GVFS (Gnome VFS is now deprecated) libraries. It supports a full undo and redoes system as well as search and replace. Other typical code oriented features include line numbering, bracket matching, text wrapping, current line highlighting, automatic indentation and automatic file backup. Some advanced features of gedit include Multilanguage spellchecking and a flexible plug-in system allowing to dynamically adding new features.

ADVANTAGES OF NS2

- Cheap- Does not require costly equipment
- Complex scenarios can be easily tested.
- Results can be quickly obtained – more ideas can be tested in a smaller time frame.
- Supported protocols
- Supported platforms
- Modularity

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 PROPOSED MODULES

- Wireless Network Initialization and MANET Deployment
- Self-Organization Phase
- Scheduling Key Computing Phase
- Operational Phase
- SET Protocol
- Exception Handling Phase

6.1.1 NETWORK INITIALIZATION AND MANET DEPLOYMENT

In this module, a wireless network is created. All the nodes are configured and randomly deployed in the network area. Since our network is a wireless network, nodes are assigned with mobility (movement). A routing protocol is implemented in the network. Sender and receiver nodes are randomly selected and the communication is initiated. All the nodes are configured to SGKP and reverse tracking among all the nodes.

6.1.2 SELF-ORGANIZATION PHASE

- After random deployment of the sensor nodes in the sensor field, the self-organization phase starts.
- It is the first phase of the protocol. During this phase, the clusters are formed. From 4 CH, the current CH and the two inter cluster nodes(sender & gateway) are selected by the BS.
- Initially, the BS collects the current location information from each of the sensor nodes and then forms a sensor field map also generate the key.
- Input
  - Specifically, randomly select two large primes p and q, and let indicate input data.
  - Each participant is an entity and denoted as Ui. Moreover, all participants have the same computation power.
- Output
  - The participant in the group or cluster is represented with hU1,U2,..Uni, where n is the number of participants in the group.

6.1.3 SCHEDULING KEY COMPUTING PHASE

- The sensor nodes can be in either of the two states active and dormant.
- Some sensor nodes are scheduled for public key state, and private key state.
- A node in generate master key neither any sensing task nor any relaying task.
- This approach is operated based on the observation that if two senor nodes are in close proximity, then there is a very high probability that they sense similar and redundant data from the environment.

- Generate an encryption key k for the homomorphic encryption scheme to encrypt data messages, where k, is a large integer.
- Generate the pairing parameters p, q. Select a generator key stochastically.
- Input
  - Randomly selects two short-term secret keys
- Output
  - Uj broadcasts (ωj ,Aj ,Bj ,M) to other participant in the cluster.

6.1.4 OPERATIONAL PHASE

- During this phase, actual key data transmissions take place.
- The sensor nodes forward data toward the CH node according to their respective medium access time slots.
- The CH nodes remove the redundancies in the data sent by the sensor nodes by the process of data aggregation and finally forward the aggregated data toward the BS as per the communication pattern distributed by the BS.
- Node j first obtains its private key as setk from mk and ID, where ID is node ID, and t is the time stamp of node j's time interval in the current round that is generated by its CH i from the network control.
- Input
  - Upon receiving the message, each sensor node verifies the authenticity in the following way.
  - It checks the time stamp of current time interval t and determines whether the received message is fresh.
- Output

- If all of the equations hold for any participant Uk, then Uj sets verification matrix value as Uj, k = “success”.

6.1.5 SET PROTOCOL

- The proposed Secure and Efficient data Transmission protocol, called SET using key generation scheme, respectively.
- Input
  - Each participant Uj checks for the message Vi m = “failure” for Um, where m 6= i.
- Output
  - If no malicious participant is detected, each Ui * Ck calculates the key for each Ci = {U1, U2, Um}

6.1.6 EXCEPTION HANDLING PHASE

- This phase is an occasional one. Due to the node mobility and the sudden death of some sensor nodes, the CH node may lose enough links with its cluster members.
- This may significantly degrade the throughput level in terms of packet delivery at the BS.
- Under this situation, the BS may send feedback to the CH, and the CH then checks the current connectivity with its cluster members.
- If there is significant loss of connectivity with its cluster members, then the CH is asked to relinquish the charge of cluster headship, and a new one is

selected either from the CH panel or one from within the two nodes already selected.

- Input
  - After the cluster keys are calculated, the merging operation is realized to produce a group key for larger group. LetC1,C2,...,Cs be the clusters in MANET then the clusters are merged
- Output
  - After the clusters have been merged, the non-clustered participants join the group. On the other hand, this step is also used for adding new participants after the group key is computed.
  - Otherwise, Ui set verification matrix value as Vj, k = “failure” and cancel the cluster key computation.

6.2 PERFORMANCE ANALYSIS

In this module, the performance of the proposed network coding method is analyzed. Based on the analyzed results X-graphs are plotted. Throughput, delay, energy consumption are the basic parameters considered here and X-graphs are plotted for these parameters.

Finally, the results obtained from this module is compared with previous results and comparison X-graphs are plotted. Form the comparison result, final RESULT is concluded.

PERFORMANCE METRICS

We have used following performance metrics for evaluating effects of attack and effectiveness of our detection algorithm:

Throughput:

It is the ratio of the total number of bits transmitted (Btx) to the time required for this transmission, i.e. the difference of data transmission end time and start time (tstart).

$$\text{Throughput} = (B_{tx})/(t_{end} - t_{start}) \text{ bps}$$

Packet Delivery Ratio:

This is defined as the ratio of the number of packets received at the destination and the number of packets sent by the source. Here, $pktd_i$ is the number of packets received by the destination node in the ith application, and $pkts_i$ is the number of packets sent by the source node in the ith application.

Average End-to-End Delay:

It is average transmission delay of packets transmitted from source to destination. D is computed as the ratio of the sum of individual delay of each received data packet to the total number of data packets received.

D= no.of received packed/total time

6.3 RESULT AND DISCUSSION

In this section, we will present the performance analysis of TCP-SACK under various variants of attack over WSNs. Simulation results using random waypoint are obtained on various scenarios, each scenario investigated for changes in performance metrics with increase in number of -attackers and hop-length (i.e. intermediate nodes) en-route. In this work, we have used two WSN scenarios for

evaluating performance. Number of nodes are statically placed in a row, end-points being source and destination nodes. Intermediate nodes along this row are as per requisite number of hops and nodes en-route. This does not take into account the complexities of the environment interference and wireless channel characteristics. Objective is to analyze the precise impact of node(s) on TCP performance. WSN consists of 40 nodes placed randomly in an area of 600 m square. This scenario is used to verify the effectiveness of proposed detection mechanism in large scale WSNs.

In our experiments, the source and destination pairs as well as attacker positions are selected randomly. All the simulation results presented are an average of ten different simulations (randomized using 10 different seeds) to offset any setup bias. Table shows the simulation parameters for aforementioned WSN scenarios.

In SGKP-WSN, each participant in the group executes Cluster Head Selection Step to divide the group U into clusters. If any malicious attempt of a participant is detected during the execution of this step, the owner of the malicious attempt is removed from the group. Once the clusters are formed, each participant in each cluster executes Temporary Public Key Distribution Step to compute and distribute the temporary public keys to the other participants in its cluster. Then, each participant in the cluster executes Temporary Public Key Verification and Secret Key Distribution Step for verifying the incoming temporary public keys, computing and broadcasting its own secret key to other participants in its own cluster. Next, secret keys are verified in Secret Key Verification Step. If any malicious attempt is detected either in temporary public key verification or in secret key verification, the owner of the malicious attempt is excluded from the cluster key computation. Each honest participant in the cluster computes the cluster

key. After the cluster keys are computed, Cluster Merge Step is executed to compute a common key for all clusters in the network. Finally, Joining Non-Clustered Participants Step is executed if there exist any non-clustered participants in the network. The resulting key is denoted as a group key for a WSN. Moreover, Leaving Participants Step can be executed if any participant leaves the group to update the group key. Details of SGKP-WSN protocol steps are as follows:

Cluster Head Selection

Each node and participants randomly selects and broadcasts the RREQ-Route Request to the adjacent participants in its network. Each participant verifies the incoming broadcast messages of each RREQ. Then, each node generates its own ACK RREP to the adjacency matrix by verifying incoming broadcast messages.

After the verification, if no cheating participant is detected, the participant with the maximum adjacent node in its neighbourhood is selected as the cluster head. In case of equality, the participant with the minimum ID is selected as the cluster head. For instance, let RREP the order are the participants with the maximum adjacent node in their neighbourhood. Then, high value node is selected as the cluster head. In our project, we have used 40 nodes which consists of 4 Cluster heads and each Cluster heads has 9 cluster members.

Public Key Distribution and Verification

Each node randomly selects two short-term secret prime key, and its broadcasts to other participant in the cluster. After the public keys are distributed, each node, where each node denotes cluster number for some positive integer l, verifies the broadcast messages for each node.

After the broadcast messages at RREQ and RREP are exchanged by node in the cluster, each node verifies the broadcast messages for each node in the cluster. Otherwise, node set verification matrix value is not equal – it will “failure” and repeat.

RSA Cluster Key Computation

If no malicious (no failure) participant is detected, then calculates the key private for each node. After the cluster keys are calculated, the data transmission operation is realized to produce a master key for larger group. Let be the clusters in WSN then the clusters are ready to transmit the data securely.

After the clusters data in transmission, the non-clustered participants join the group. On the other hand, this step is also used for adding new participants after the group key is computed. Let be the participant set after the cluster verification step and the non-clustered node can send data or act an intermediate nodes.

Table-I

SIMULATION PARAMETERS

Parameter	Value
Simulator	NS2
Simulation time	10s
Area	1400X1400
Number of node	40
Physical Layer	IEEE 802.11
Routing protocol	AODV
Mobility model	Random way point
Radio type	802.11a/g
Transmission rate	10 packets/s
Packet Size	512/ 1024
Pause time	0s

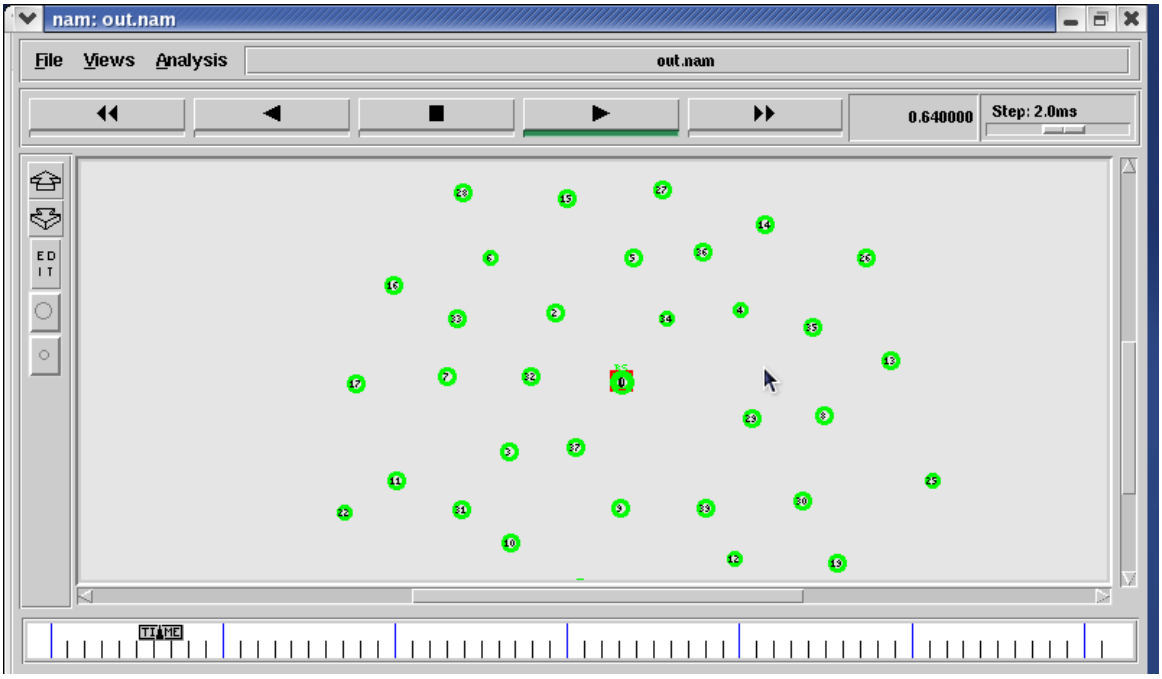


Fig.6.1 Network Deployment

Simulation results are shown in Figures. Full active CSMA/CA has the smallest delay for all traffic load, other three MAC's latencies increase significantly when the traffic load is larger than a certain threshold.

The output is displayed step by step. Firstly 40 WSN nodes are arranged. we have taken node 0 as Base Station and it is represented as square shape.

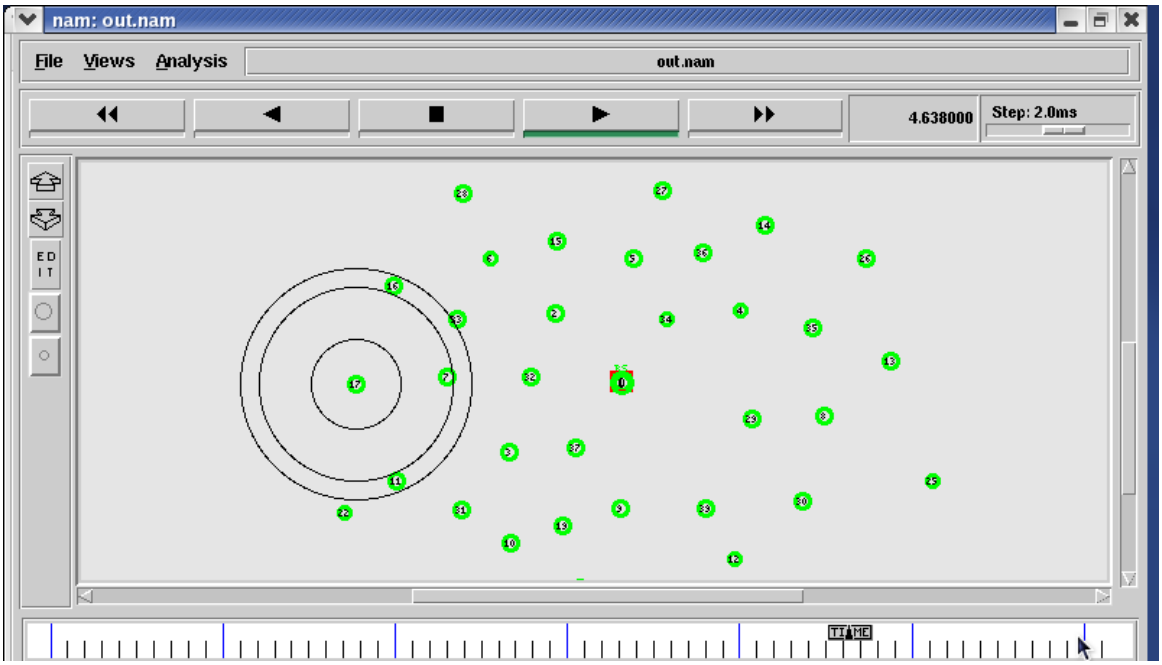


Fig.6.2 Clustering Process Scenario1

In Fig.6.2 ,The Clustering process is initiated. Generally, Cluster consists of a cluster head and remaining nodes are cluster members, which communicates only with cluster heads. The main goal of our project is to reduce the energy consumption and delay.

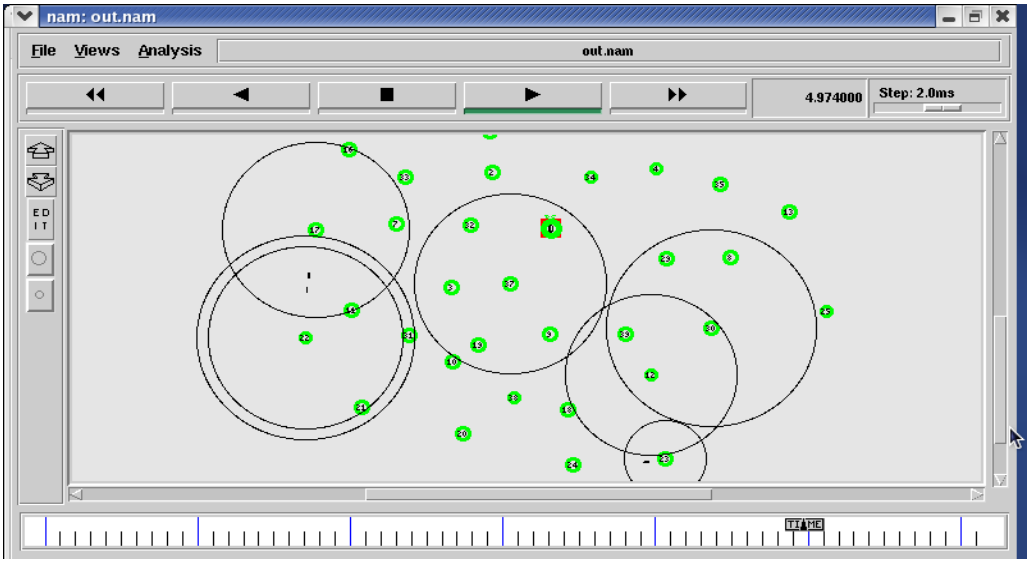


Fig.6.3 Data Transmission Scenario 2

40 nodes are randomly placed in a 1400m×1400m area. All sources generate traffic at one message per 3 seconds. Sources and destination nodes are randomly selected.

PERFORMANCE ANALYSIS

In this project, the performance of the proposed SGKP based security method is analyzed. The parameter such as Throughput, delay, energy consumption and X-graphs are plotted. Finally, the results obtained from the proposed system is compared with existing system.

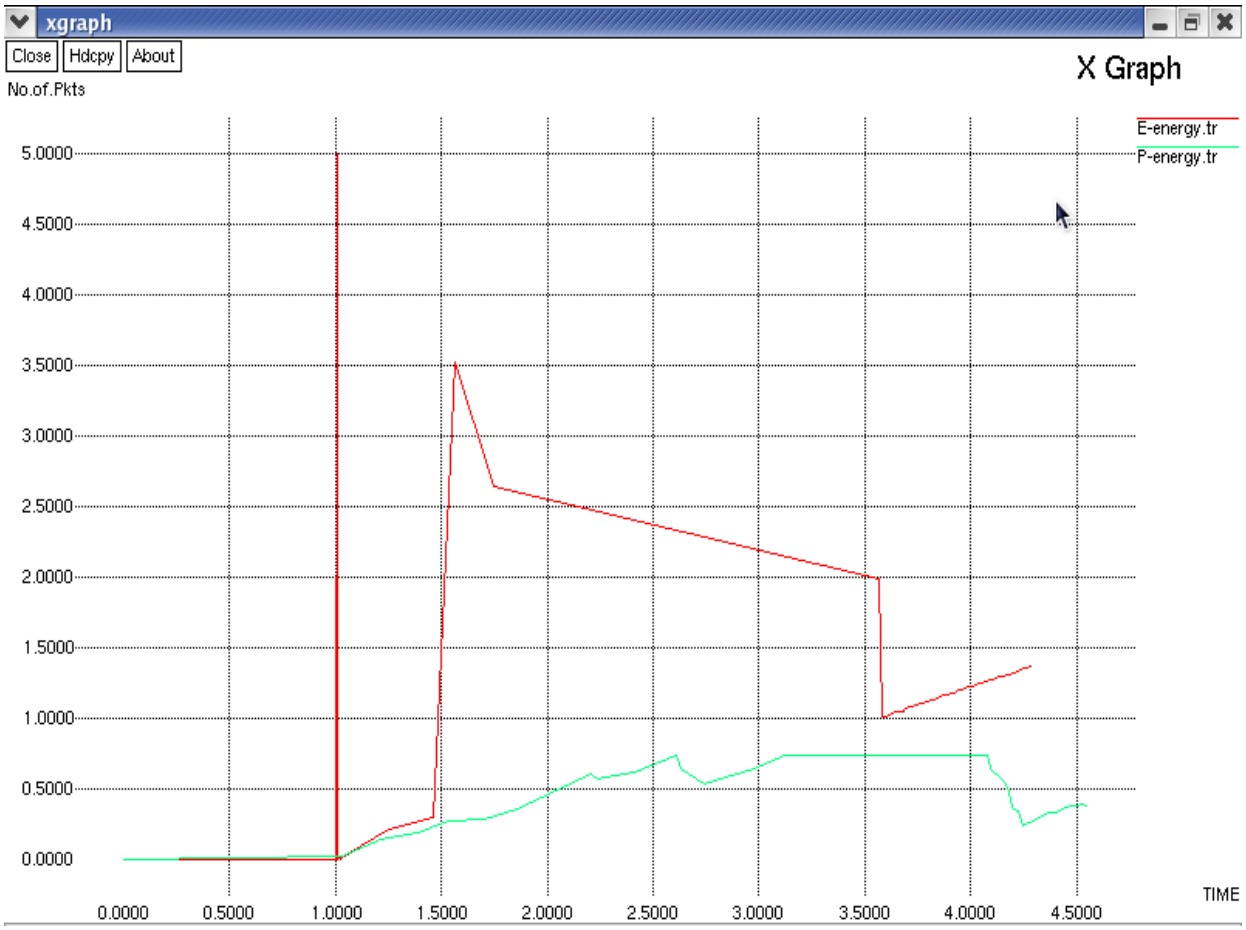


Figure.6.4 Energy vs. No. Of Nodes

The Figure shows the utilization of energy in the SET proposed topology compared with Signature based intrusion detection system. The red line represents the existing system's energy consumption and green line represents the proposed system's energy consumption. In Signature based intrusion detection system, the Attack detection period is always available so the graph increases gradually and hence consumes more energy. But in SET protocol, we are using RSA Algorithm to generate keys. After generating the keys, attack detection period takes place. So the energy consumption in SET protocol is reduced .

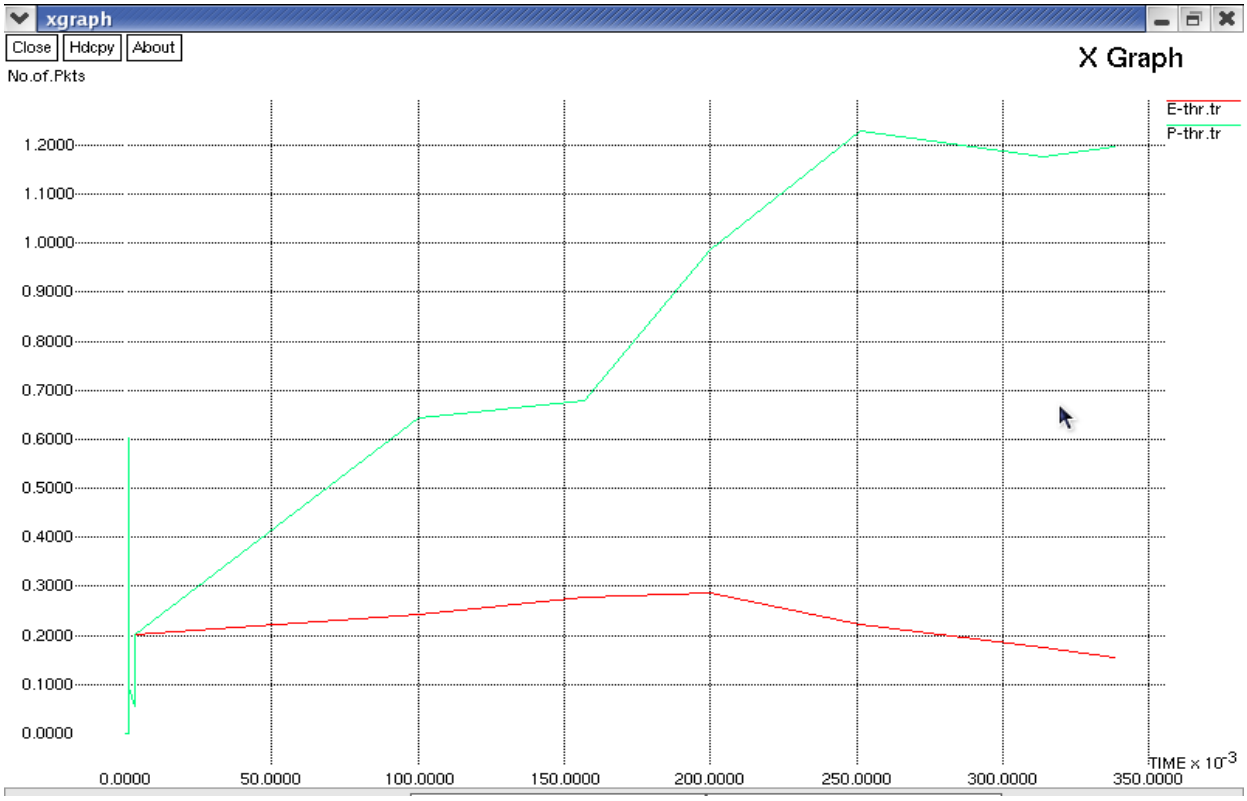


Figure.6.5 Throughput vs. No. Of. Nodes

The red line represents the existing system's throughput and green line represents the proposed system's throughput. The throughput in proposed system is increased compared to existing system, since Signature based intrusion detection system takes place in existing system. We noticed that SET protocol has given an average of 34% improvement in the network throughput.

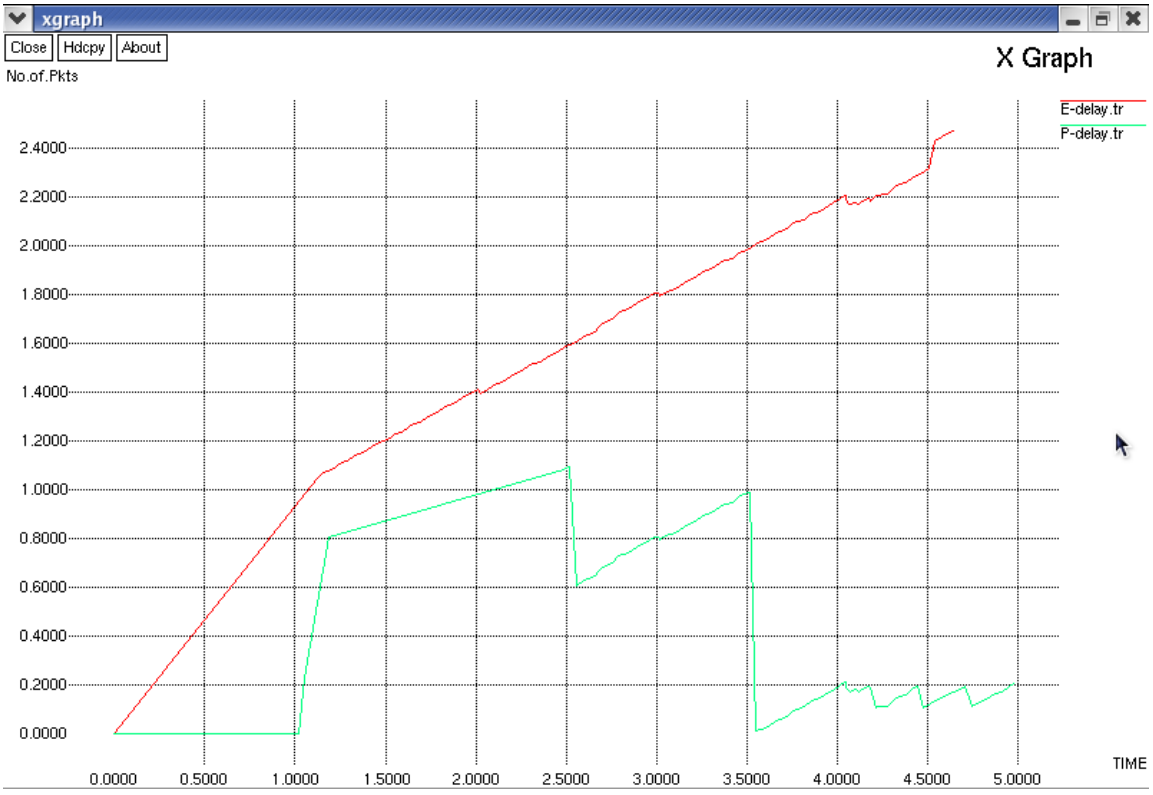


Figure.6.6Delay vs. Nodes

As shown in Figure, the proposed SET-SGKP scheme generates lower normalized routing overhead than the existing and proposed routing protocol. The red line represents the existing system's delay and green line represents the proposed system's delay. The graph increases gradually in existing system since Signature based intrusion detection system takes place and various steps are used to detect the malicous nodes. We are using SET protocol in proposed system , the graph increases only in attack detection period and decreases after key generation.

CHAPTER 7

CONCLUSION

In this work, we design a highly efficient and secure WSN protocol, to collect information from a large number of sensors. Since we are using SET Protocol, it reduced Energy Consumption, Time Delay, Throughput compared to our existing system. This WSN protocol is specified by a dense set of wireless sensors in the network, out of which a unknown subset may be active and transmit at the same time to a sink. We then presented secure and efficient data transmission protocols, respectively, for WSN network. Furthermore, a novel and secure cluster head selection mechanism has been proposed. Our analyses show that SET has better performance results than the existing ones in terms of the communications and the computational costs. For the computational cost analysis, only the modular exponentiation operations of key computation steps have been considered. In short, the performance of independent of the number of participants in the group for both communications cost and the computational cost. Simulation results show that the proposed SET protocol have better performance than existing secure protocols.

To validate the efficiency, of both the non-secure and secure WSN models suggested, we provide rigorous analysis, numerical results and simulations, which illustrate how the different parameters of the network affect the performance of the protocol, namely, the rate of the wireless channel.

FUTURE ENHANCEMENT

Future enhancement would be integrating advanced encryption algorithms specially designed for resource constrained environment, ensuring even stronger security without sacrificing efficiency. Implementing key management techniques could enhance security by regularly updating encryption keys to mitigate the risk of key compromise.

APPENDIX- I

CODING

```
set val(chan)          Channel/WirelessChannel
set val(prop)          Propagation/TwoRayGround
set val(netif)         Phy/WirelessPhy
set val(mac)           Mac/802_11
set val(ifq)           Queue/DropTail
set val(ll)            LL
set val(ant)           Antenna/OmniAntenna
set val(ifqlen)        1000
set val(nn)            40
set val(rp)            AODV
set val(x)             1400
set val(y)             1400
set val(mob)           "location1"

puts "This is a multi-channel sensor network test program."

set ns_          [new Simulator]
set tracefd [open out.tr w]
$ns_ trace-all $tracefd
```

```
set namtrace [open out.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set topo   [new Topography]

$topoload_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
set chan_2_ [new $val(chan)]

set val(rp) PHENOM                      ;
$ns_ node-config \
    -adhocRouting $val(rp) \
        -llType $val(ll) \
        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -antType $val(ant) \
        -propType $val(prop) \
        -phyType $val(netif) \
        -channel $chan_1_ \
        -topoInstance $topo \
```

```
            -energyModelEnergyModel \
            -initialEnergy 100 \
            -rxPower 12 \
            -txPower 20 \
            -agentTrace ON \
            -routerTrace ON \
            -macTrace ON \
            -movementTrace ON

 set node_(0) [$ns_ node 0]

 $god_ new_node $node_(0)
    $node_(0) namattach $namtrace
    $ns_ initial_node_pos $node_(0) 30
    [$node_(0) set ragent_] pulserate .05      ;
    [$node_(0) set ragent_] phenomenon CO      ;

set val(rp) AODV                          ;
$ns_ node-config \
     -adhocRouting $val(rp) \
            -channel $chan_2_ \
              -PHENOMchannel $chan_1_
```

```
for {set i 1} {$i<40 } {incri} {
        set node_($i) [$ns_ node]
        $god_ new_node $node_($i)
   $node_($i) namattach $namtrace
   $ns_ initial_node_pos $node_($i) 20
}

set sensor1 [new Agent/SensorAgent]
$ns_ attach-agent $node_(0) $sensor1
[$node_(1) set ll_(1)] up-target $sensor1
set src [new Agent/UDP]
set sink [new Agent/UDP]
$ns_ attach-agent $node_(1) $src
$ns_ attach-agent $node_(0) $sink
source $val(mob)

source link.tcl

$ns_ at 6.1 "stop"
$ns_ at 6.11 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
   global ns_ tracefdnamtrace
```

```
  $ns_ flush-trace
    close $tracefd
    close $namtrace
}

puts "Starting Simulation..."
$ns_ run
```

REFERENCES

[1] E.Tsukerman, ‘‘How machine learning is revolutionizing intrusion detec tion,’’ in Designing a Machine Learning Intrusion Detection System. Berkeley, CA, USA: Apress, 2020,

[2] Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, ‘‘Cyber intrusion detection by combined feature selection algorithm,’’ J. Inf. Secur. Appl., vol. 44, pp. 80–88, Feb. 2019.

[3] S. Boubiche, D. E. Boubiche, A. Bilami, and H. Toral-Cruz, “Big data challenges and data aggregation strategies in wireless sensor networks,” IEEE Access, vol. 6, pp. 20558–20571, 2018.

[4] P. Li, W. Zhao, Q. Liu, X. Liu, and L. Yu, ‘‘Poisoning machine learning based wireless IDSs via stealing learning model,’’ in Proc. Int. Conf. Wireless Algorithms, Syst., Appl., 2018

[5] A. De Bonis and U. Vaccaro, “-almost selectors and their applications to multiple-access communication,” IEEE Trans. Inf. Theory, vol. 63, no. 11, pp. 7304–7319, Nov. 2017.

[6] C. Aksoylar, G. K. Atia, and V. Saligrama, “Sparse signal processing with linear and nonlinear observations: A unified Shannon-theoretic approach,” IEEE Trans. Inf. Theory, vol. 63, no. 2, pp. 749–776, Feb. 2017

[7] A.Ghosal and S. Halder, ‘‘A survey on energy efficient intrusion detection in wireless sensor networks,’’ J. Ambient Intell. Smart Environ., vol. 9, no. 2, pp. 239–261, Feb. 2017

[8] C.-J. Liu, P. Huang, and L. Xiao, “TAS-MAC: A traffic-adaptive synchronous MAC protocol for wireless sensor networks,” ACM Trans. Sensor Netw., vol. 12, no. 1, p. 1, 2016.

[9] I.Almomani, B. Al-Kasasbeh, and M. AL-Akhras, ‘‘WSN-DS: A dataset for intrusion detection systems in wireless sensor networks,’’ J. Sensors, vol. 2016, pp. 1–16, Aug. 2016

[10] S. Wu, S. Wei, Y. Wang, R. Vaidyanathan, and J. Yuan, “Partition information and its transmission over boolean multi-access channels,” IEEE Trans. Inf. Theory, vol. 61, no. 2, pp. 1010–1027, Feb. 2015.

[11] M. L. Malloy and R. D. Nowak, “Near-optimal adaptive compressed sensing,” IEEE Trans. Inf. Theory, vol. 60, no. 7, pp. 4001–4012, Jul. 2014

[12] V. Y. F. Tan and G. K. Atia, “Strong impossibility results for sparse signal processing,” IEEE Signal Process. Lett., vol. 21, no. 3, pp. 260–264, Mar. 2014

[13] C. Lam Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, “Non-adaptive group testing: Explicit bounds and novel algorithms,” IEEE Trans. Inf. Theory, vol. 60, no. 5, pp. 3019–3035, May 2014.

[14] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, “The evolution of MAC protocols in wireless sensor networks: A survey,” IEEE Communication Surveys Tuts., vol. 15, no. 1, pp. 101–120, 1st Quart., 2013.

[15] G. K. Atia and V. Saligrama, “Boolean compressed sensing and noisy group testing,” IEEE Trans. Inf. Theory, vol. 58, no. 3, pp. 1880–1901, Mar. 2012