Target Tracking in Wireless Sensor Network

Author

Umair Shafiq Khan

NUST201363009MCEME35213F

Supervisor

Dr. Muazzam A. Khan

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

SEPTEMBER, 2016

Target Tracking in Wireless Sensor Network

Author

UMAIR SHAFIQ KHAN

NUST201363009MCEME35213F

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Computer Engineering

Thesis Supervisor:

Dr. Muazzam A. Khan

Thesis Supervisor's Signature: ____

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

SEPTEMBER, 2016

Declaration

I certify that this research work titled "*Target Tracking in Wireless Sensor Network"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

UMAIR SHAFIQ KHAN

NUST201363009MCEME35213F

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

UMAIR SHAFIQ KHAN

NUST201363009MCEME35213F

Signature of Supervisor

Copyright Statement

Acknowledgements

In the name of ALLAH, who is most merciful, the one and only supreme power. After Allah, I would like to thanks my parents who really worked hard and guided me in the right direction. They always advised me to look forward and move onward to fulfill my dreams. My parent's prayers and efforts have made me able to achieve the goals they dream for me. I really want to thank them from the core of my heart.

I wish to express my sincere thanks and gratitude to my advisor Dr. Muazzam A. Khan for his sincere guidance, constructive comments, hard work and efforts. I am much impressed by his way of teaching, especially the way he guided me for my research work. Without his motivation, encouragement and help, this project would have never reached to its completion.

I also want to express my thanks to the faculty of Department of Computer Engineering specially our respected professors for their teachings in the courses I studied and the knowledge I was able to gain from them.

Dedicated to my exceptional parents, adored siblings and my wife whose tremendous support and cooperation led me to this wonderful accomplishment

Abstract

With increasing advancement and popularity of wireless sensor networks, their applications in different areas are becoming more and more sophisticated. As wireless sensor networks and its components are distributed in nature, easy to configure, flexible to deploy and less costly, they are considered feasible to collect, process and store data from different terrains. Amongst multiple applications of wireless sensor networks, tracking a particular object is one of the most important ones. While tracking an object, its geographical location is determined and then constantly monitored. Every sensor in the wireless sensor network participates in sensing the position of the object being tracked, and shares this information with other components of the network. In recent years much work has been done regarding target tracking in wireless sensor networks and many algorithms have been proposed to perform the task most efficiently and optimally. In this thesis a simple approach is used to track a target which significantly improves the accuracy and other network performance parameters by minimizing the communication overhead.

Key Words: Object Tracking, Wireless Sensor Network, NS2

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 WIRELESS SENSOR NETWORKS:

A wireless sensor network is a bunch of customized transducers with architecture capable to communicate, used for monitoring and recording different physical and environmental conditions at diverse locations. Usually observed values are temperature, humidity, pressure, wind direction and speed, illumination intensity, vibration intensity, sound intensity, power-line voltage, chemical concentrations, pollutant levels and vital body functions [21].

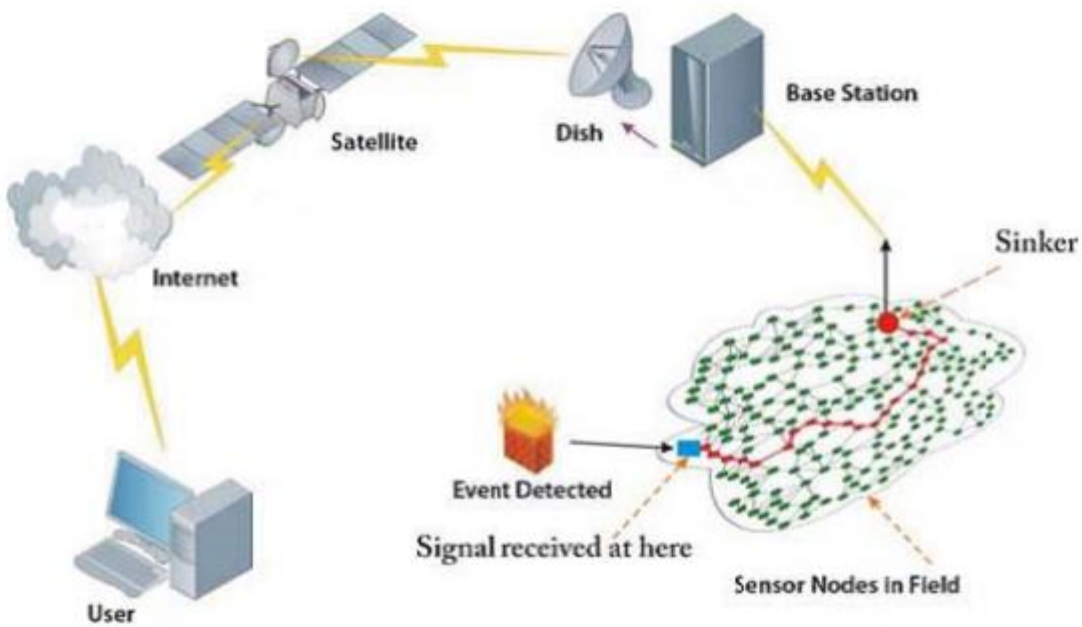


Figure 1: A Typical WSN [10]

A sensor network contains many different detection devices known as sensor nodes, which are tiny, lightweight and portable. Basic components of every mote are a sensor or actuator, microcontroller, transceiver with internal or external antenna and a power source. The transducer throws electrical signals after sensing conditions around it. The microcontroller on board of a sensor processes and buffers the values given by sensor. The transceiver’s job is to receive instructions from a central entity and to send data to that device. The power supply for each sensor node is provided by a battery.

1.1.1 Characteristics of WSN:

- Provide a scaffold between the actual physical and virtual world.
- Allow the capacity to watch the beforehand inconspicuous at a fine determination over vast spatio-transient scales
- Have an extensive variety of potential applications to industry, science, transportation, common foundation, and security.

- **Less expensive:** In the wireless sensor network regularly thousands or a huge number of motes are conveyed to quantify the physical or environmental conditions. With a specific end goal to diminish the general expense of the entire system the expense of mote should be minimum.
- **Energy proficient:** Vitality in WSN is utilized for various reason, for example, calculation, correspondence and capacity. Sensor motes devours more vitality contrast with some other for correspondence. In the event that they come up short on the force they regularly get to be invalid as we don't have any alternative to energize. Thus, the conventions and calculation advancement ought to consider the force utilization in the outline stage.
- **Processing Capabilities:** Typically, the mote has limited processing capability by considering the other expenses.
- **Communication Capabilities:** Wireless sensor network normally uses the wireless channel through radio waves for communication. They can communicate each other with limited transmission capacity. The communication channel can be either bidirectional or unidirectional. Unattended and threatening environment can be hard to run wireless sensor network easily. Along these lines, the equipment and programming for correspondence must need to prolong the communication.
- **Security and Privacy:** Each mote ought to have adequate security systems so as to avert unapproved access, assaults, and inadvertent harm of the data within the sensor hub. Moreover, extra security instruments should likewise be incorporated. Appropriated detecting and preparing: the extensive number of sensor hub is circulated consistently or haphazardly. WSNs every hub is equipped for gathering, sorting, preparing, accumulating and sending the information to the sink. Consequently, the conveyed detecting gives the vigor of the framework.
- **Dynamic system topology:** The sensor hub can fall flat for battery depletion or different circumstances, correspondence channel can be disturbed and additionally the extra sensor hub

might be added to the system that outcome the continuous changes in the system topology. Along these lines, the WSN hubs must be installed with the capacity of reconfiguration, self-conformity.

- **Self-association:** The sensor hubs in the system must have the ability of sorting out themselves as the sensor hubs are conveyed in an obscure manner in an unattended and threatening environment. The sensor hubs have work in coordinated effort to change themselves to the conveyed calculation and structure the system naturally.
- **Multi-hop correspondence:** An expansive number of sensor hubs are sent in WSN. In this way, the achievable approach to speak with the sinker or base station is to take the assistance of a halfway hub through directing way. On the off chance that one has to speak with the other hub or base station which is past its radio recurrence it must me through the multi-jump course by halfway hub.
- **Application oriented:** WSN is unique in relation to the customary system because of its tendency. It is profoundly reliant on the application ranges from military, ecological and also wellbeing segment. The hubs are sent haphazardly and spread over relying upon the kind of utilization.
- **Robust Operations:** Since the sensors will be conveyed over a substantial and once in a while threatening environment. Along these lines, the sensor hubs must be shortcoming and blunder tolerant. Accordingly, sensor hubs require the capacity to individual test, self-adjust, and self-repair
- **Small physical size:** Sensor nodes are little in size with the limited reach. Because of its size its vitality is restricted which makes the correspondence ability low.

1.1.2 Architecture of WSN:

WSN is changing which can comprise of different sorts of sensor hubs. The earth is heterogeneous as far as both equipment and additionally programming. The sensor hub development centers to lessen cost, build adaptability, give adaptation to non-critical failure. Enhance advancement process and save vitality. The structure of sensor hub comprises of detecting unit (sensor and simple to advanced converter), preparing unit (processor and capacity), correspondence unit (handset), and force supply unit. The significant pieces appeared in Fig. 2 a compact depiction of various unit is as per the following:

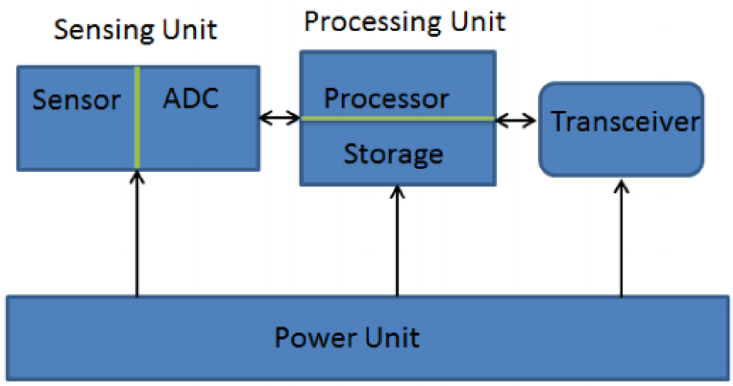


Figure 2: Structure of a Sensor Node [9]

Detecting unit: it is made out of accumulation of various sorts of sensor which is required for estimation of various marvel of the physical environment. Sensors are chosen in view of its application. Sensor out is electric sign which is simple. Along these lines, simple to-advanced converter (ADC) is utilized to change the sign to computerized to speak with the microcontroller.

Handling unit: it comprises of a processor (microcontroller) and capacity (RAM). Likewise, it has working frameworks and in addition clock. The obligation of the preparing unit incorporates gathering information from different sources than handling and putting away. Clock is utilized to do the sequencing for the grouping.

Correspondence unit: it utilizes a handset which comprises of a transmitter and in addition a collector. The correspondence is performed through the correspondence channels by utilizing the system convention. Taking into account the application necessities and importance so as to impart it regularly utilizes reasonable strategy, for example, radio, infrared or optical correspondence.

Power unit: the assignment of the force unit is to give the vitality to the sensor hub for checking the earth easily and less time. The life of the sensor relies on upon the battery or force generator which is associated with the force unit. Power unit is required for the productive utilization of the battery.

1.1.3 WSN communication architecture:

WSN communication architecture is somewhat not quite the same as the ordinary PC correspondence and PC system. The correspondence engineering can be arranged in various layers. So as to get the most extreme productivity with restricted assets and low overhead WSN does not hold fast as nearly to the layered engineering of OSI model of traditional system. By and by, the layered model is helpful in WSN for classifying conventions, assaults and barrier. Thus, rather than the conventional seven layers it is lessened to the five layers [2] that incorporate physical layer, Data join layer, system layer, transport layer and application later. The benefit of the layered model is adroitly comparative capacities are consolidated at one layer. Fig. 3 demonstrates the correspondence convention model of remote sensor system.

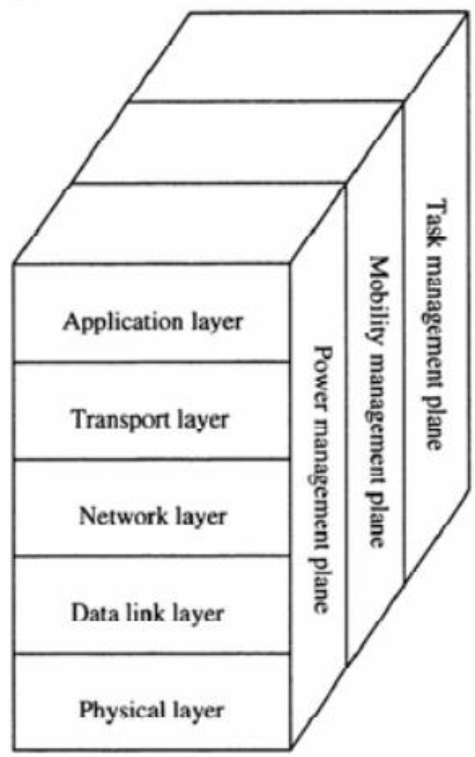


Figure 3: Protocol Stack of WSN [11]

The physical layer addresses the equipment point of interest of remote correspondence system. This layer is in charge of recurrence determination, bearer recurrence era, signal location, adjustment, and information encryption. The information join layer is worried with the media access control (MAC) convention. Since the remote channel is vulnerable to the clamor and sensor hubs might change the area MAC convention at the information join layer must be force mindful and ought to have the capacity of minimizing the collisions [12].

The system layer deals with the steering the information supplied by the vehicle layer or between the hubs. Though the transport layer can keep up the information stream if the WSN application requires that. Different sort of use can be actualized in the application depending the physical natural detecting.

1.1.4 WSN applications:

WSN can screen distinctive physical qualities: temperature, moistness, light, weight, commotion, soil organization, object movement (discovery, and following), objects weight, size, and so forth. The sensors additionally can transmit and forward detecting information to the base station. Most present day WSNs are bi-directional, empowering two-way correspondence, which could gather detecting information from sensors to the base station and additionally spread charges from base station to end sensors. The advancement of WSNs was persuaded by military applications, for example, front line reconnaissance; WSNs are generally utilized as a part of modern situations, private situations and natural life situations. Structure wellbeing checking, medicinal services applications, home computerization, and creature following are a portion of the agent WSNs applications. Various uses of WSNs in various fields are:

Environmental monitoring: There are numerous applications in checking ecological parameters like Air contamination observing, Forest flame location, Landslide recognition, Water quality checking, and Natural debacle aversion. Fig. 4 indicates Environmental observing. In this different sensors are spread over the trees in the woodland zones. These sensors report to the current climate detecting station and the temperature of the backwoods is accounted for to the climate detecting station which ceaselessly speaks with the satellite and the satellite is associated with the flame checking station. As the temperature exceeds a particular edge esteem, the control focuses are cautioned and essential move is made to give assistance to the required spot.

Acoustic identification: is the study of utilizing sound to decide the separation and bearing of something. Area should be possible effectively or latently, and can happen in gasses, fluids, and in solids. Dynamic acoustic area includes the production of sound with a specific end goal to create a reverberation, which is then dissected to decide the area of the article being referred to. Aloof acoustic area includes the discovery of sound or vibration made by the article being recognized, which is then broke down to decide the area of the item being referred to.

Seismic Detection: Seismic waves produced by blasts or vibrating controlled sources are one of the essential strategies for underground investigation. Controlled-source seismology has been utilized to guide salt arches, anticlines and other geologic traps in petroleum bearing rocks, flaws, rock sorts, and since quite a while ago covered mammoth meteor pits. Seismometers are sensors that sense and record the movement of the Earth emerging from versatile waves. Seismometers might be sent at the Earth's surface, in shallow vaults, in boreholes, or submerged.

Medical monitoring: The medicinal applications can be of two sorts: wearable and embedded. Wearable gadgets are utilized on the body surface of a human. The implantable therapeutic gadgets are those that are embedded inside human body. Body-region organizes in this way shaped can gather data from wearable and embedded therapeutic gadgets around an individual's wellbeing, wellness, and vitality consumption. Figure 5 demonstrates the part of remote sensor systems in therapeutic applications. In this there is indoor observing and in addition open air checking which is finished with the assistance of versatile stations and base station. The data from the base station is sent to the IP Network which goes about as a portal from the checking side to the side. Subsequent to getting the data about the patient with the assistance of IP Network, therapeutic help is given to the patient.

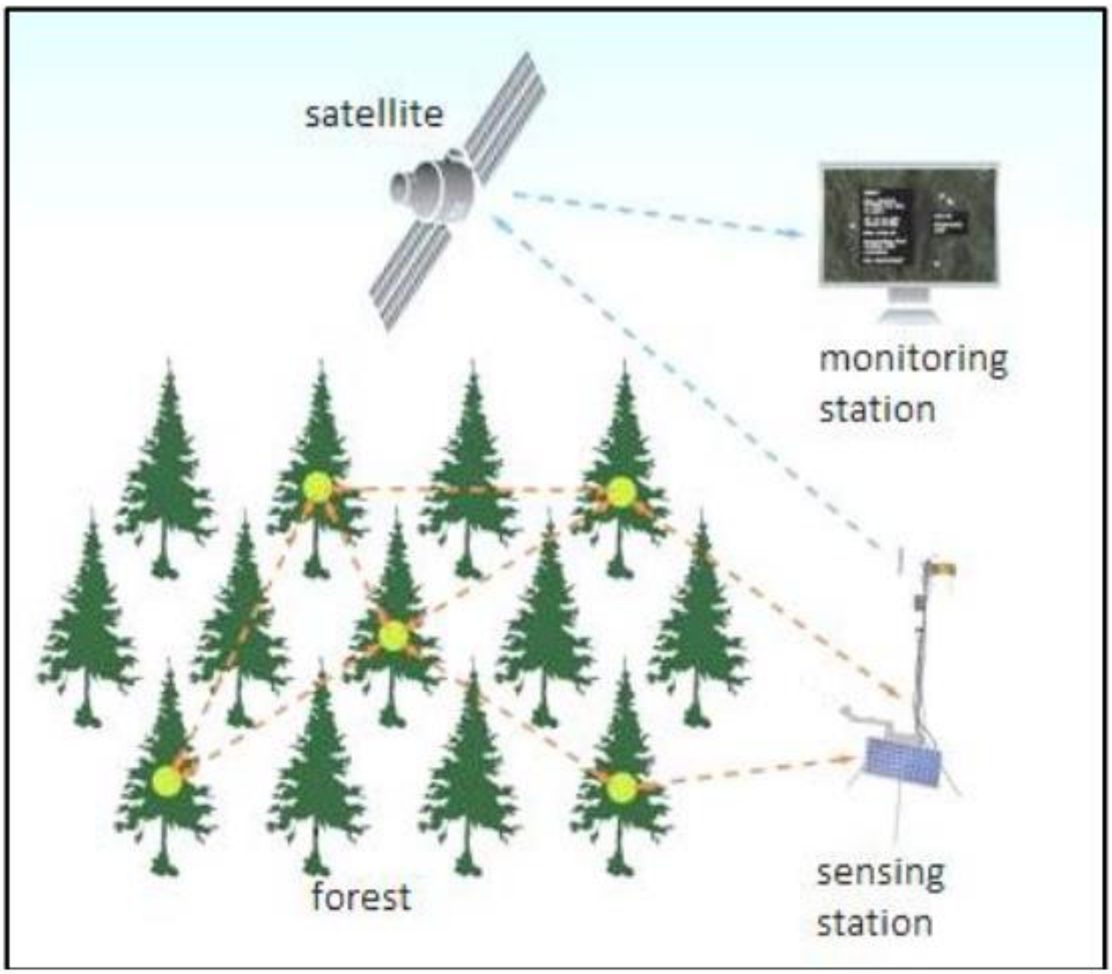


Figure 4: Environmental Monitoring using WSN [8]

Security and Surveillance: The center of observation missions is to obtain and check data about adversary abilities and positions of antagonistic targets. Such missions frequently include a high component of danger for human staff. Henceforth, the capacity to send unmanned reconnaissance missions, by utilizing remote sensor systems, is of incredible down to earth significance for the military. In any case, because of the vitality imperatives of sensor gadgets, such frameworks require a vitality mindful outline to guarantee the life span of observation missions. The framework permits a gathering of collaborating sensor gadgets to distinguish and track the positions of moving vehicles in a vitality proficient and stealthy way.

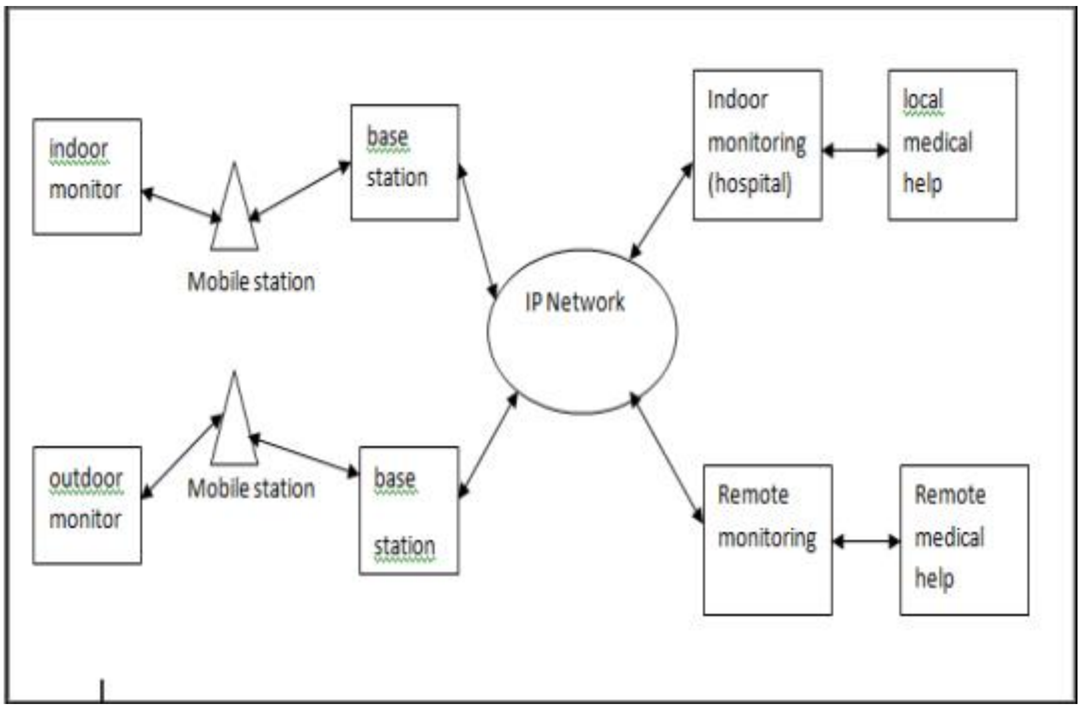


Figure 5: Remote Sensor systems in Therapeutic Applications [8]

For convenience in interoperability and implementations of new technologies in new fields of communications and information technology, defining standards is a must. When it's about VANETs, all the functionalities of physical layer to application layer are specified after standardization. Usually in literature the complete protocol stack of VANETs is designated by DSRC, WAVE etc. which are explained as follows;

1.2 TARGET TRACKING IN WSN:

Target following (appeared in Fig. 6) is a standout amongst the most imperative applications in WSN in which sensor hubs screen and report the positions of moving items to the application's client with a base inertness.

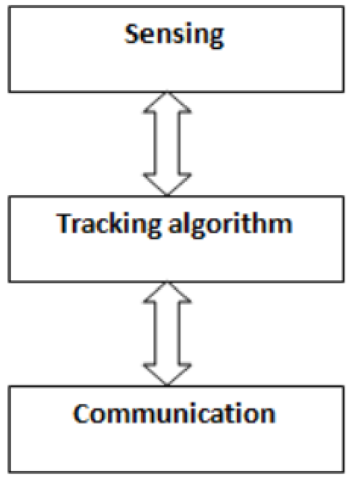


Figure 6: Target Tracking Scheme Classification

Truth be told, target following has numerous genuine applications, for example, war zone observation, recognition of illicit fringes crossing, gas spillage, fire spread, and untamed life checking.

Target following can be performed utilizing a solitary hub or through the coordinated effort between various sensors. Notwithstanding, utilizing a solitary hub may bring about force misfortune and affects overwhelming calculation trouble on that hub, while utilizing numerous sensors gives better results as far as exactness and vitality sparing, because of the collaboration between hubs. Target following is the use of WSN whose objective is to follow the wandering way of an item which is considered as an objective and to distinguish the position of target. As WSN persistently screen nature, it gives us space to upgrade the vitality productivity. Target following plan includes three interrelated subsystems which are appeared in the figure 6.

The detecting subsystem is utilized to sense the objective i.e. it involves the hub that to start with identifies the objective and different hubs which steadily participate in recognizing the target. Second subsystem is the forecast based calculation which is utilized to follow the way of the fancied target. The last one is correspondence subsystem which is utilized to send the data starting with one hub then onto the next. All these three subsystem works cooperatively and keeps up the relationship among them.

With remote sensor systems, articles can be followed by essentially labeling them with a little sensor hub. The sensor hub will be followed as it travels through a field of sensor hubs that are conveyed in the earth at known areas. Rather than detecting ecological information, these hubs will be conveyed to sense the RF messages of the hubs connected to different articles. The hubs can be utilized as dynamic labels that report the nearness of a gadget. A database can be utilized to record the area of followed articles with respect to the arrangement of hubs at known areas. With this framework, it gets to be conceivable to ask where an item is at present, not just where it was last checked. Not at all like detecting or security systems, hub following applications will consistently have topology changes as hubs travel through the system. While the availability between the hubs at settled areas will remain generally steady, the network to portable hubs will be constantly evolving. Moreover, the arrangement of hubs being followed will consistently change as articles enter and leave the framework. It is fundamental that the system have the capacity to proficiently identify the nearness of new hubs that enter the system.

1.2.1 Localization and target tracking challenges:

Numerous difficulties can influence the objective following quality in remote sensor systems:

Node disappointment: nodes in WSNs are inclined to disappointments because of battery weariness, physical catastrophes, equipment disappointments, outside assaults, et cetera. Subsequently, proposed target following conventions ought to adapt to these difficulties.

Target missing and recuperation: forecast blunders, obstructions, sudden changes in target direction, and pace cause loss of target. To handle this test, vigorous following calculations ought to be proposed to diminish the likelihood of missing targets. In addition, recuperation systems must be considered in the event that the objective was lost.

Coverage and availability: there is a shared relationship amongst scope and target following. High scope results in high following precision. Be that as it may, the exhibitions of target following calculations are debased on account of inadequate systems, or within the sight of scope gaps.

Data accumulation: the total component intends to take out information repetition. In a bunched system, the hubs exchange their detected information to their related group head, which performs information total and dispenses with redundancies. Along these lines, the information conglomeration ought to be expert in a precise way, with least information idleness while protecting vitality.

Tracking dormancy: the execution of target following calculations must be performed quickly while protecting situating exactness. At the point when the operation of following takes too long, the moving hub may change its area.

Energy utilization: since sensor hubs keep running on batteries which are, as a rule, nonrechargeable, vitality proficiency is a basic issue in remote sensor arranges particularly in touchy applications like target following.

CHAPTER 2: LITERATURE REVIEW

In WSNs Moving article following has gotten Considerable consideration as of late and the arrangements can be mostly characterized into five plans, for example, tree - based following, bunch based following; forecast based following; mobicast message-based following and half breed techniques and Actuation based techniques.

2.1 Tree-Based Target Tracking:

In Tree-Based Target Tracking, hubs in a system might be sorted out in a various leveled tree structure or spoke to as a chart. The vertices in the structure speak to sensor hubs and edges are connections between hubs that can specifically speak with each other. While following an objective the hubs that identify the objective speaks with each other also, chooses a root hub. The root hub gathers data from every one of the hubs by means of a circulated spanning tree. In the event that the root hub is far from the objective, then the tree will be reconfigured. In spite of the fact that the crossing tree based methodologies track the moving items all the more precisely, tree associations result in high-vitality utilizations [29]. The unified target following methodologies are both time and vitality devouring; to evade this constraint tree-based tracking techniques are proposed [26].

2.2 CLUSTER BASED:

Cluster based strategy isolates the system into groups to bolster communitarian information preparing. A bunch comprises of group head and part sensor hubs. At the point when a sensor identifies an article it volunteers to go about as a CH. There is No need of express race of pioneer. So message trades are not brought about. In the event that more than one effective sensor may identify the sign, different volunteer hubs may exist. So a decentralized methodology must be connected to guarantee that as it were one Cluster Head is dynamic in the region of an objective to be followed with high likelihood. Bunch based technique is separated into three sorts. Static Clustering, Dynamic Clustering and Space Time Clustering. Static bunching is separated into Prediction based grouping and Non Forecast based Clustering. Dynamic Clustering comprises of a few strategies like IDSQ [23], DELTA [24], and RARE [25]. DSTC is one of the Space time bunching strategy.

2.3 HYBRID METHOD:

Hybrid techniques are the following calculations that satisfy the prerequisites of more than one write of target following. Half and half Clustering comprises of a few techniques like DPT, DCAT, and Various leveled forecast system (HPS).

In [1], author proposed another technique for target following that makes adjustment with target portability model. This strategy uses two instruments to make flexibility which are changing the size and state of bunches as per target versatility model. Additionally, by consolidating static and element bunching a semi dynamic grouping structure has been created to execute these instruments. Reproduction results demonstrate that our recommended strategy diminishes both vitality utilization by diminishing bunches size when the objective moves consistently, and following mistake by changing the size and the state of groups as per target versatility when the objective moves unusually.

This paper [1] addresses object following issues in light of the voracious technique proposing a productive following strategy, alluded to as the "Large Frequency First Tree (LFFT)". The proposed technique is intended to track a substantial number of moving items by productively gathering and conglomerating all following data. It additionally proposes the Tree Adaptation Procedure (TAP) to lessen the item following tree overhauling costs. It likewise proposes another area determination arrangement, alluded to as the "focal point of gravity approach". The reproduction results demonstrate that the overhaul cost for certain tree development calculations can be enhanced utilizing the focal point of gravity approach. The redesign cost for the article following tree will be decreased when TAP is performed. The reproduction results demonstrate that LFFT accomplishes great execution when TAP is connected once

This paper [3] presents a vitality proficient bunching calculation for versatile sensor system in view of the LEACH convention. The proposed convention adds highlight to LEACH to bolster for versatile hubs furthermore diminishes the utilization of the system asset in each round. The proposed convention is reproduced and the outcomes demonstrate a noteworthy diminishment in system vitality utilization contrasted with LEACH [20].

This study [4] devises and assesses a vitality effective circulated cooperative sign and data preparing system for acoustic target following in remote sensor systems. The disseminated preparing calculation depends on portable specialist processing worldview and consecutive Bayesian estimation. At every time step, the short identification reports of group individuals will be gathered by bunch head, and a sensor hub with the most astounding sign to-commotion proportion (SNR) is picked there as reference hub for time distinction of arrive (TDOA) figuring. Amid the portable operator relocation, the objective state conviction is transmitted among hubs and upgraded utilizing the TDOA estimation of these combination hubs one by one. The registering and preparing weight is uniformly appropriated in the sensor system. To diminish the remote interchanges, we propose to speak to the conviction by parameterized strategies, for example, Gaussian estimation or Gaussian blend model guess. Moreover, we show a fascination power capacity to handle the versatile operator relocation arranging issue, which is a mix of the hub remaining vitality, helpful data, and correspondence cost. Recreation illustrations show the estimation adequacy and vitality proficiency of the proposed conveyed community target following structure.

Authors in [5] present the idea of element guard tree-based cooperation, and formalize it as a various target improvement issue which needs to discover a caravan tree grouping with high tree scope and low vitality utilization. They propose an ideal arrangement which accomplishes 100% scope and minimizes the vitality utilization under certain perfect circumstances. Considering the genuine limitations of a sensor system, we propose a few pragmatic usages: the preservationist plan and the expectation based plan for tree extension and pruning; the consecutive and the restricted reconfiguration plans for tree reconfiguration. Broad trials are led to think about the functional usage and the ideal arrangement. The outcomes demonstrate that the forecast based plan beats the preservationist plan and it can accomplish comparable scope and vitality utilization to the ideal arrangement. The analyses likewise demonstrate that the confined reconfiguration plan beats the successive reconfiguration plan when the hub thickness is high, and the pattern is turned around when the hub thickness is low.

In this [6] paper, creator propose and assess a circulated, vitality productive, light-weight structure for target confinement and following in remote sensor systems. Since radio correspondence is the most vitality devouring operation, this structure intends to diminish the quantity of messages and the quantity of message crashes, while giving refined precision. The key component of the system is a novel confinement calculation, called Ratio Metric Vector Iteration (RVI). RVI depends on separation proportion assesses instead of supreme separation gauges which are frequently difficult to ascertain. By iteratively overhauling the evaluated area utilizing the separation proportion, RVI restricts the objective precisely with just three sensors' interest. After restriction, the area of the objective is accounted for to the supporter. In the event that the objective is stationary or moves around inside a little zone, it is inefficient to report (nearly) the same area evaluates over and over. Creator in this way, propose to powerfully modify a reporting recurrence considering the objective's development so we can decrease the quantity of report messages while keeping up following quality. Broad recreation results demonstrate that the proposed system consolidating RVI and the development versatile report planning calculation lessens the restriction mistake and aggregate number of the transmitted messages up to half of those of the current methodologies.

The methodology in [7] depends on a face steering and forecast technique. We utilize a state move system, a dynamic vitality utilization model, and a moving target situating model to diminish vitality utilization by requiring just a base number of sensor hubs to take an interest in correspondence, exchange, and detecting for target following. Two sensor hubs, in particular, "Screen" and 'Reinforcement', are utilized for target following for every timeframe. For the entire time of target following, a connected rundown of screen and reinforcement sensors is shaped. On the off chance that either screen or reinforcement sensor comes up short, this methodology can in any case survive. Recreation results contrasted and existing conventions indicate better following exactness, quicker target catching pace, and better vitality effectiveness.

CHAPTER 3: NETWORK SIMULATOR (Version 2)

NS2 is an open source discrete event tool [27], [28] primarily developed for the research in wired and wireless communication systems. It supports a large number of protocols required for the networking simulations i.e. TCP, UDP, Routing and multicasting.

NS2 development started as the variant of REAL Network simulator which was originally developed by the department of Electrical engineering and computer science at University of California, Berkeley in 1989. Later on, Defense Advanced Research Projects Agency (DARPA) supported the development through VINT project at LBL, Xerox PARC, UCB, and USC/ISI and currently collaborating through the SAMAN and NSF with CONSER. [13] Many individual researchers are the part of NS2 community and continuously contributing in the development since its inception. Although NS2 is an industry leader and extensively used for the simulation of computer networks, there are many bugs reported and recovered from time to time which are helping it to make a strong platform for the researchers.

3.1 Architecture:

As shown in Figure 7, NS2 comprises on two basic languages C++ and OTCL. Simulation objects are created in these two languages and TclCL is used to link these languages together. TCL Simulation Script is used to call those simulation objects through an executable command of ns. After running the script, it generates a Simulation Trace File which can be used for the animation and plotting to view the output.

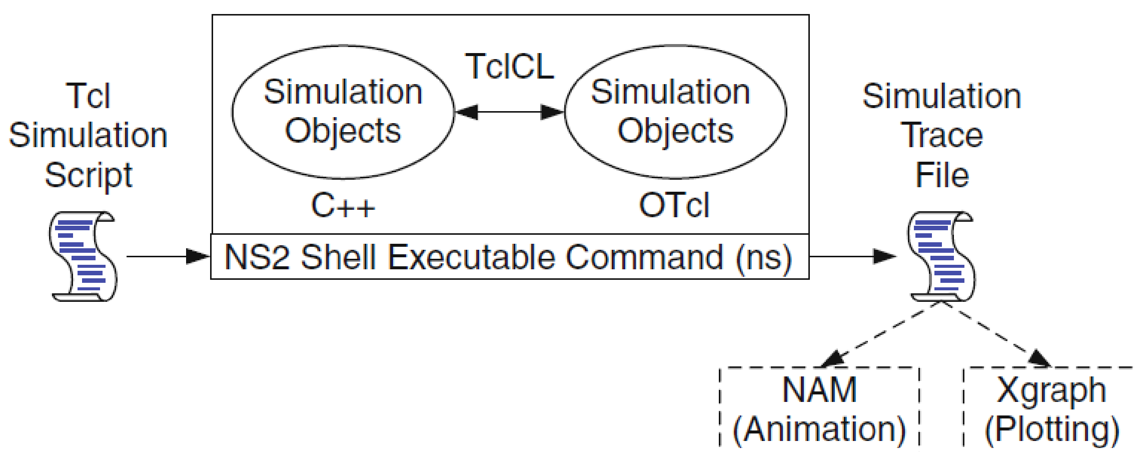


Figure 7: Basic Architecture of NS [14]

3.2 Installation:

NS2 is an open source simulator which can be downloaded from the official site of ISI. [13] It is compatible with most of the Unix and Linux based Operating systems. However, Cygwin is required for its installation on Windows based system. The current ns-allinone-2.35 package as shown in Figure 8 includes

- NS 2.35
- Tcl/TK 8.5.10
- OTcl 1.14
- TclCl 1.20
- NAM 1.15
- Xgraph 12.2

Before installation development tools i.e. gcc, gcc-g++ are required as a pre-requisite.

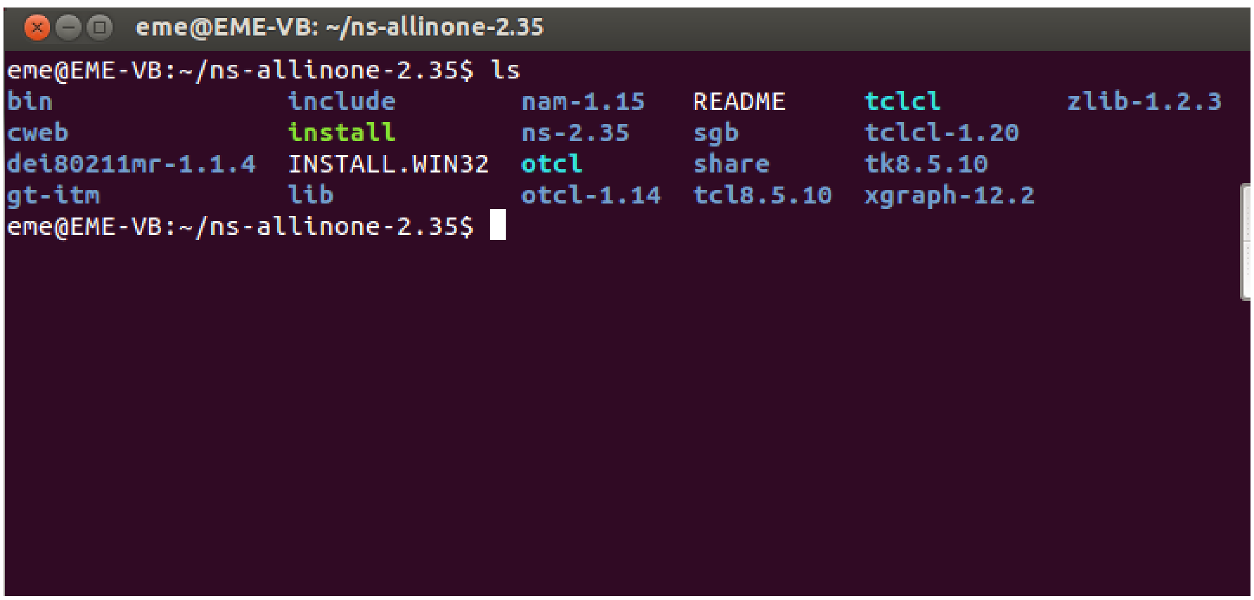


Figure 8: ns-allinone-2.35 on Ubuntu 12.02

3.3 TCL Scripting:

To write a script in TCL for the simulation purposes, steps are given below;

1 Preamble

2 Predefine Tracing

3 Define the topology and links

4 Define Agents to carry traffic

5 Attach applications with agents

6 Finish procedure

7 Start simulation

For better understanding, let's start with an example [15]:

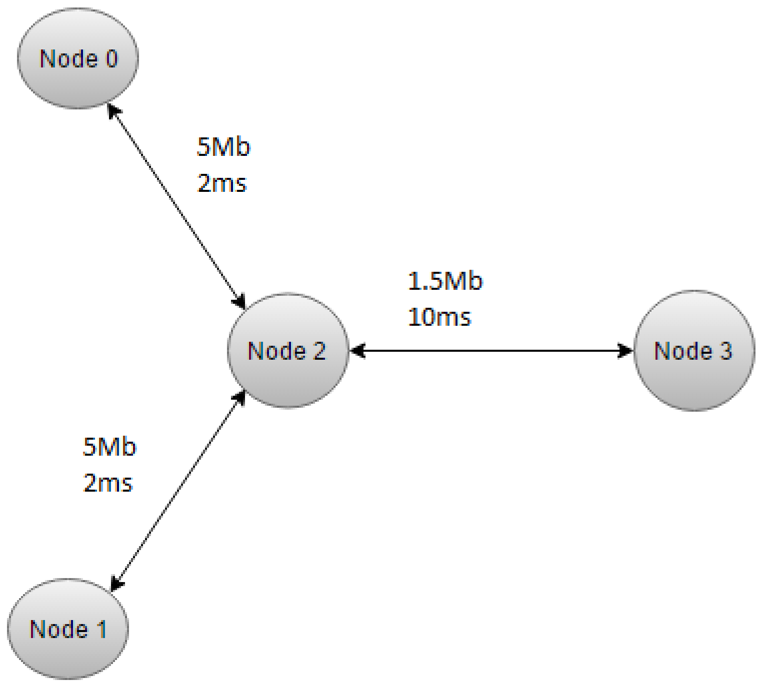


Figure 9: Network Topology

1. **The preamble**

   set ns [new Simulator]

2. **Predefine tracing**

   set f [open out.tr w]

   $ns trace-all $f

   set nf [open out.nam w]

   $ns namtrace-all $nf

3. **Define the topology and links as shown in Figure. 9**

   set node0 [$ns node]

   set node1 [$ns node]

```
set node2 [$ns node]
set node3 [$ns node]
$ns duplex-link $node0 $node2 5Mb 2ms DropTail
$ns duplex-link $node1 $node2 5Mb 2ms DropTail
$ns duplex-link $node2 $node3 1.5Mb 10ms DropTail
```

4. **Define Agents to carry traffic**

```
set udp0 [new Agent/UDP]
$ns attach-agent $node0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$udp0 set class_ 0
set null0 [new Agent/Null]
$ns attach-agent $node3 $null0
$ns connect $udp0 $null0
$ns at 1.0 "$cbr0 start"
puts [$cbr0 set packetSize_]
puts [$cbr0 set interval_]
```

5. **Attach application with Agents**

```
set tcp [new Agent/TCP]
$tcp set class_ 1
$ns attach-agent $node1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $node3 $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.2 "$ftp start"
$ns connect $tcp $sink
$ns at 1.35 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"
$ns at 3.0 "finish"
```

6. **Finish Procedure**

```
proc finish {} {
```

global ns f nf

$ns flush-trace

close $f

close $nf

puts "running nam..."

exec nam out.nam &

exit 0

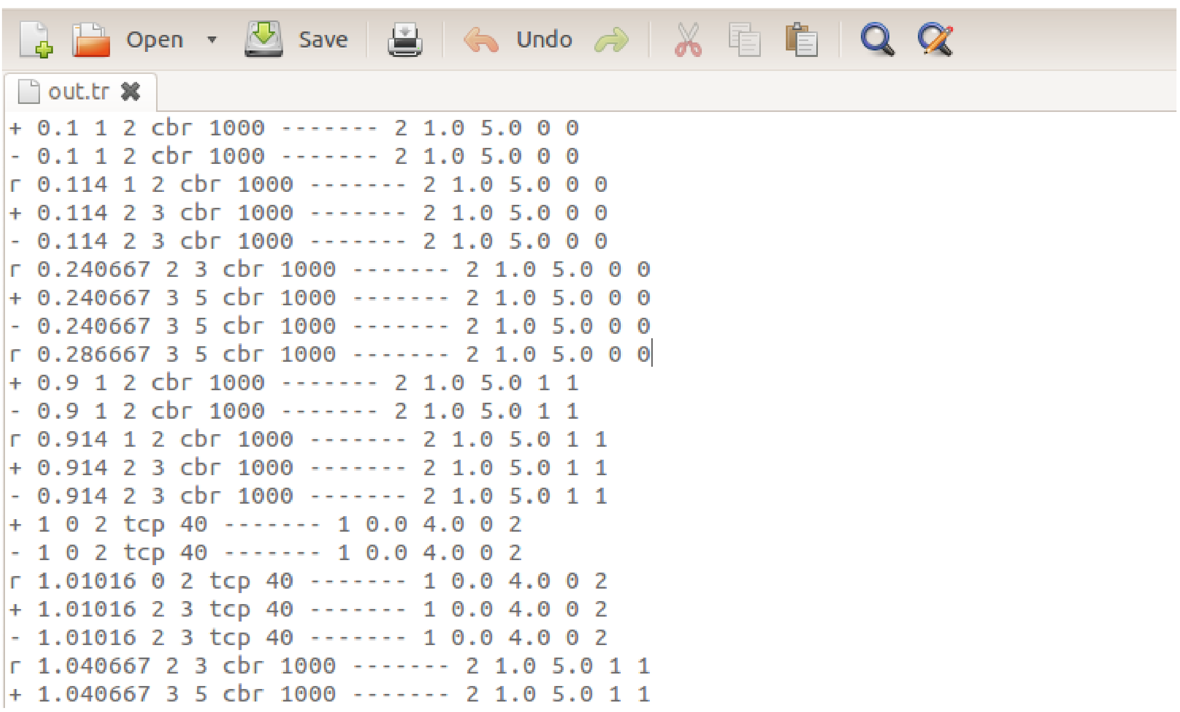
}

7. **Start simulation.**

$ns run

3.4 Trace File:

The file written by the NS2 which contains all the necessary information about the network and records all the events during the simulation. Figure [10] and Figure [11] shows the trace file formats of wired and wireless networks respectively.




```
+ 0.1 1 2 cbr 1000 ------- 2 1.0 5.0 0 0
- 0.1 1 2 cbr 1000 ------- 2 1.0 5.0 0 0
r 0.114 1 2 cbr 1000 ------- 2 1.0 5.0 0 0
+ 0.114 2 3 cbr 1000 ------- 2 1.0 5.0 0 0
- 0.114 2 3 cbr 1000 ------- 2 1.0 5.0 0 0
r 0.240667 2 3 cbr 1000 ------- 2 1.0 5.0 0 0
+ 0.240667 3 5 cbr 1000 ------- 2 1.0 5.0 0 0
- 0.240667 3 5 cbr 1000 ------- 2 1.0 5.0 0 0
r 0.286667 3 5 cbr 1000 ------- 2 1.0 5.0 0 0
+ 0.9 1 2 cbr 1000 ------- 2 1.0 5.0 1 1
- 0.9 1 2 cbr 1000 ------- 2 1.0 5.0 1 1
r 0.914 1 2 cbr 1000 ------- 2 1.0 5.0 1 1
+ 0.914 2 3 cbr 1000 ------- 2 1.0 5.0 1 1
- 0.914 2 3 cbr 1000 ------- 2 1.0 5.0 1 1
+ 1 0 2 tcp 40 ------- 1 0.0 4.0 0 2
- 1 0 2 tcp 40 ------- 1 0.0 4.0 0 2
r 1.01016 0 2 tcp 40 ------- 1 0.0 4.0 0 2
+ 1.01016 2 3 tcp 40 ------- 1 0.0 4.0 0 2
- 1.01016 2 3 tcp 40 ------- 1 0.0 4.0 0 2
r 1.040667 2 3 cbr 1000 ------- 2 1.0 5.0 1 1
+ 1.040667 3 5 cbr 1000 ------- 2 1.0 5.0 1 1
```

Figure 10: Trace file of Wired Network

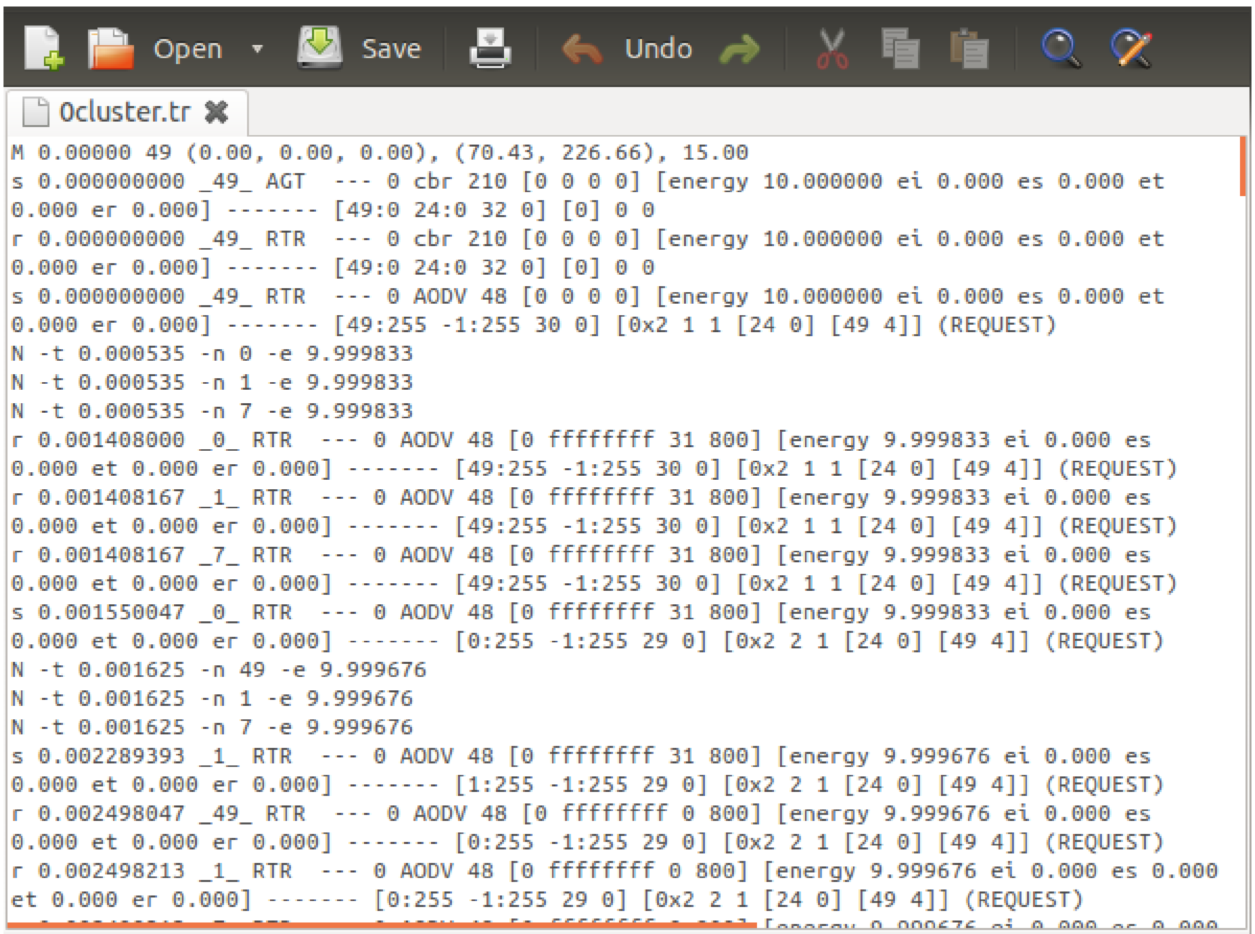
The format of trace file for wired network is shown below:

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

Figure 11: Trace File Format

There are 12 fields and according to ns2 manual, we can explain these fields as

1. 1^{st} field defines the packets type
2. 2^{nd} field defines the time at which operation performed at that packet
3. 3^{rd} and 4^{th} field shows the source and destination ID's of Nodes.
4. 5^{th} field shows the packet type whether it is a CBR or RTR.
5. 6th field shows the packet size in bytes and its size increases from upper to lower layer
6. 7th field shows the flags
7. 8th field shows the flow ID
8. 9^{th} and 10^{th} field shows the source and destination addresses
9. 11^{th} field shows the sequence number for analysis purposes
10. 12^{th} field shows the packet unique id




```
M 0.00000 49 (0.00, 0.00, 0.00), (70.43, 226.66), 15.00
s 0.000000000 _49_ AGT  --- 0 cbr 210 [0 0 0 0] [energy 10.000000 ei 0.000 es 0.000 et
0.000 er 0.000] ------- [49:0 24:0 32 0] [0] 0 0
r 0.000000000 _49_ RTR  --- 0 cbr 210 [0 0 0 0] [energy 10.000000 ei 0.000 es 0.000 et
0.000 er 0.000] ------- [49:0 24:0 32 0] [0] 0 0
s 0.000000000 _49_ RTR  --- 0 AODV 48 [0 0 0 0] [energy 10.000000 ei 0.000 es 0.000 et
0.000 er 0.000] ------- [49:255 -1:255 30 0] [0x2 1 1 [24 0] [49 4]] (REQUEST)
N -t 0.000535 -n 0 -e 9.999833
N -t 0.000535 -n 1 -e 9.999833
N -t 0.000535 -n 7 -e 9.999833
r 0.001408000 _0_ RTR  --- 0 AODV 48 [0 ffffffff 31 800] [energy 9.999833 ei 0.000 es
0.000 et 0.000 er 0.000] ------- [49:255 -1:255 30 0] [0x2 1 1 [24 0] [49 4]] (REQUEST)
r 0.001408167 _1_ RTR  --- 0 AODV 48 [0 ffffffff 31 800] [energy 9.999833 ei 0.000 es
0.000 et 0.000 er 0.000] ------- [49:255 -1:255 30 0] [0x2 1 1 [24 0] [49 4]] (REQUEST)
r 0.001408167 _7_ RTR  --- 0 AODV 48 [0 ffffffff 31 800] [energy 9.999833 ei 0.000 es
0.000 et 0.000 er 0.000] ------- [49:255 -1:255 30 0] [0x2 1 1 [24 0] [49 4]] (REQUEST)
s 0.001550047 _0_ RTR  --- 0 AODV 48 [0 ffffffff 31 800] [energy 9.999833 ei 0.000 es
0.000 et 0.000 er 0.000] ------- [0:255 -1:255 29 0] [0x2 2 1 [24 0] [49 4]] (REQUEST)
N -t 0.001625 -n 49 -e 9.999676
N -t 0.001625 -n 1 -e 9.999676
N -t 0.001625 -n 7 -e 9.999676
s 0.002289393 _1_ RTR  --- 0 AODV 48 [0 ffffffff 31 800] [energy 9.999676 ei 0.000 es
0.000 et 0.000 er 0.000] ------- [1:255 -1:255 29 0] [0x2 2 1 [24 0] [49 4]] (REQUEST)
r 0.002498047 _49_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] [energy 9.999676 ei 0.000 es
0.000 et 0.000 er 0.000] ------- [0:255 -1:255 29 0] [0x2 2 1 [24 0] [49 4]] (REQUEST)
r 0.002498213 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] [energy 9.999676 ei 0.000 es 0.000
et 0.000 er 0.000] ------- [0:255 -1:255 29 0] [0x2 2 1 [24 0] [49 4]] (REQUEST)
```

Figure 12: Trace File of Wireless Network

Trace file format of Wireless Network is almost same like wired network. Although some fields are extra added for analyzing the network performance like energy model fields are described below energy: total remaining energy

ei: energy consumption in IDLE state

es: energy consumption in SLEEP state

et: energy consumed in transmitting packets

er: energy consumed in receiving packets

3.5 NAM:

NAM is an open source animation tool based on Tcl/TK to view the output of network simulator.

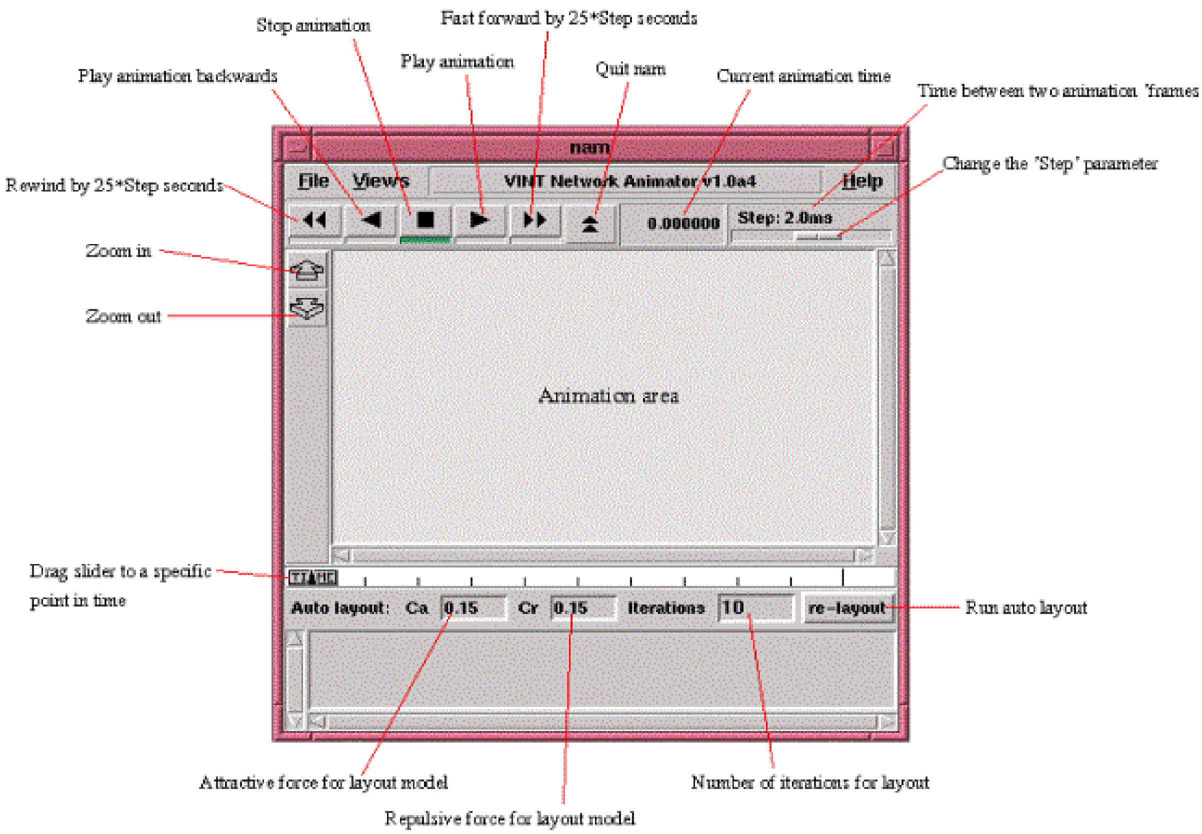


Figure 13: NAM User-interface [16]

3.6 Xgraph:

Xgraph comes along with ns-allinone package. It is a graphing utility to plot the output and analyze different characteristics of the Simulation like throughput, end to end delay etc.

For better understanding of xgraph random 5-points data is saved into a file named “Sample Data” and executed by simple command:

xgraph "Sample Data"

Sample Data

0.0 0
0.5 10
1.1 14
1.7 54
2.4 12
3.0 15

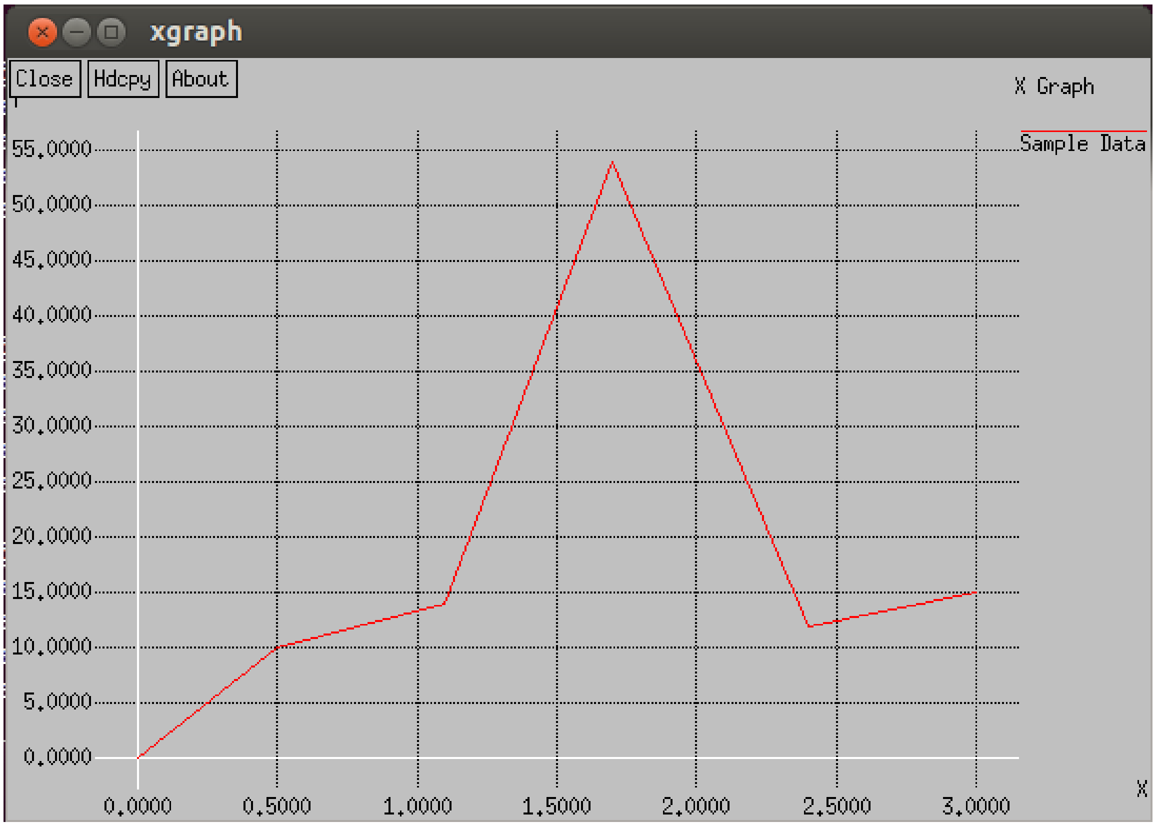


Figure 14: Plot for the Sample Data

For advance options different switches can be used in xgraph. Some of them are mentioned in the table 1 below;

Table 1: Xgraph command-line Options [19]

Options	Description
-titles	To indicate the title of graph
-x_range	To specify the x-axis range on the graph
-y_range	To specify the y-axis range on the graph
-columns	To specify the columns from the data
-pdf	For the pdf output
-fontsize_titles	To specify the font size of the titles

CHAPTER 4: AWK LANGUAGE

AWK belongs to the interpreted programming languages family and its name derived from the initials of its developers; Alfred Aho, Peter Weinberger and Brian Kernighan [22]. AWK is used to process large amount of textual data to extract required information. Due to its powerful features it is a standard language used in many Unix and Linux Based OS.

4.1 History:

Alfred Aho, Peter Weinberger and Brian Kernighan developed the AWK in 1970s while working at Bell Labs and initially it appeared in version 7 of Unix. During 1985-1988, Paul Rubin, Jay Fenlason, and Richard Stallman worked extensively on this project and released the GNU AWK in 1988 which is the most widely used version because it is the part of GNU-based linux packages.

4.2 Structure:

The basic purpose of AWK is to search the data from the text lines, for that purpose a simple code is written which search for the specific pattern and after matching the pattern it executes the action. The Basic workflow of AWK is show below in the Figure [14]. According to the one of AWK Author;
"AWK is a language for processing text files. A file is treated as a sequence of records, and by default each line is a record. Each line is broken up into a sequence of fields, so we can think of the first word in a line as the first field, the second word as the second field, and so on. An AWK program is a sequence of pattern-action statements. AWK reads the input a line at a time. A line is scanned for each pattern in the program, and for each pattern that matches, the associated action is executed." - Alfred V. Aho [17]

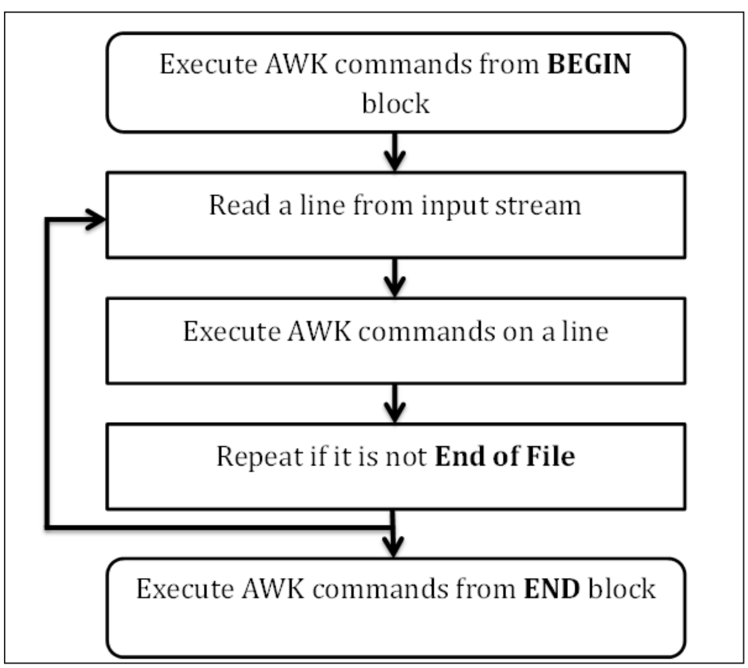


Figure 15: AWK Workflow [18]

There are two basic ways to execute the AWK commands

1. **Command line**

   by using single quote at command line
   [EME] $ awk [options] file …
   For example:
   [EME] $ awk '{print $1}' result.txt

2. **Script file**

   In this way commands are **s**aved into a file with .awk extension and accessed through the given command
   [EME] $ awk -f script.awk result.txt

4.3 Built-in Variables:

AWK provides a large number of built-in variables used for special purposes. Some of them are given below:
NF: Number of fields in the current record
NR: Number of current records
RS: Record separator
$0: Entire Input Record
$n: Nth field in the current Record

4.4 Operators:

There is a large number of operators present in the AWK i.e.

1. Arithmetic Operators (+ , - , * , / , %)
2. Increment and Decrement Operators (++ , --)
3. Assignments Operators (=)
4. Relational Operators (== , != , < , >)
5. Logical Operators (&& , || , !)

4.5 Arrays:

The good thing in AWK about arrays is that index can be any string or continuous number and the size of array can be declared later.
Syntax:
array_name[index] = value

CHAPTER 5: PROPOSED METHODOLOGY

5.1 Main Idea:

There are usually three stages on which wireless sensor network's target tracking based on; sensing, processing and communication. During all these stages, the main focus remains on the efficient, accurate and energy saving tracking mechanism. The main idea is to enhance the network life, reduce the network overhead and accurately locate the target. Figure 16 illustrates the basic concept of target tracking in WSN, first of all node sense the target and will send the information to its cluster. Some basic assumptions are necessary to made up at this point.

1. Every node has some specific sensing and communication range.
2. Network is divided on geo-based clusters
3. Every node knows its cluster head to communicate with it.
4. Every node knows its own location.
5. Every node knows about its neighbour nodes and their locations

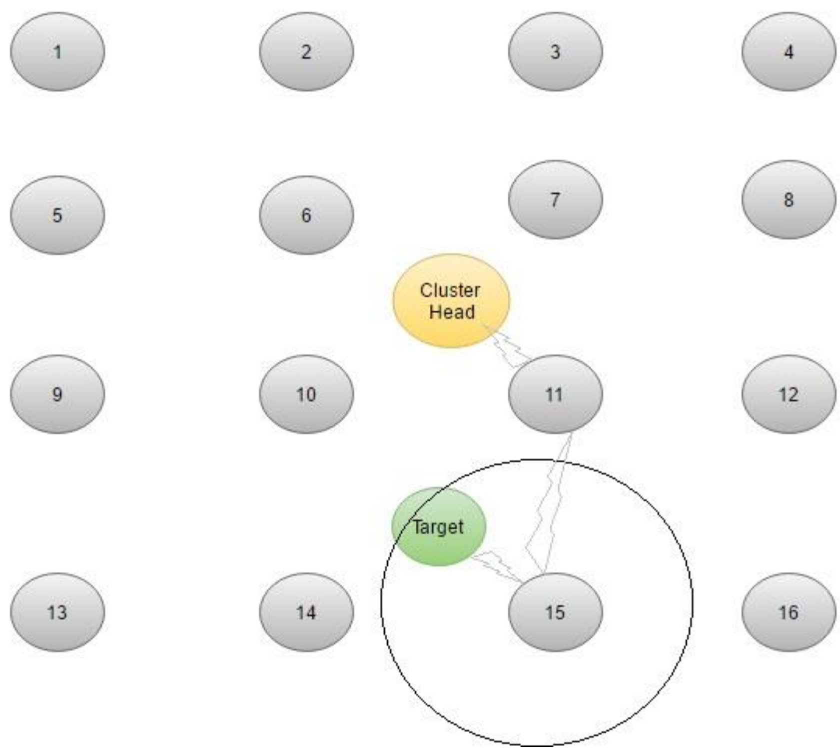


Figure 16: Target Tracking in WSN

As node sense the target, it will follow the flow chart as in Figure 17. After sensing, when cluster-head will receive the data, it will check the current node position who sense the target and calculate the distance on the basis of power received that target is how much far away from that node. After knowing the distance between target and sensing node, we know that target can be at any place around the radius of that node. So it will extract the information of node which previously sense the target. And by using simply distance formula shown in eq. given below, it will calculate the target coordinates and display the results.

$$Distance = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

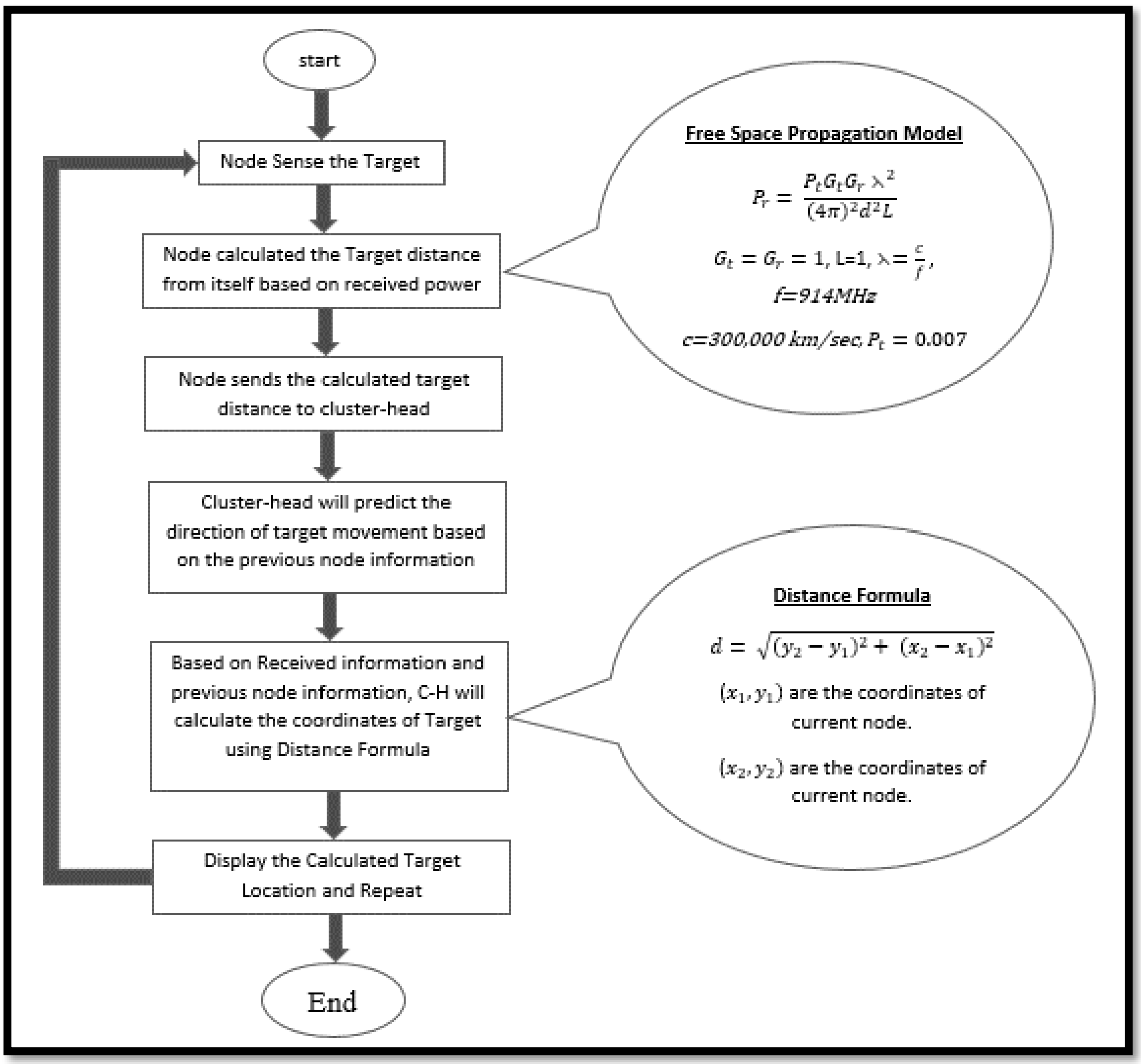


Figure 17: Flow Chart of Target Tracking in WSN

5.2 Implementation:

NS2.35, an open source platform is used to implement this technique. Simulation parameters are set as shown in table 1.

Table 2: NS2 Simulation Parameters

Parameter	**Value**
Total Nodes	52
Area	300 x 300
Channel	Wireless
Propagation Model	Free Space
Antenna	Omni
Routing Protocol	AODV
Energy Model	EnergyModel
Initial Energy	3J
Transmission Power	0.175 Watt
Receiving Power	0.175 Watt
Movement Order	Random-Way Point Model

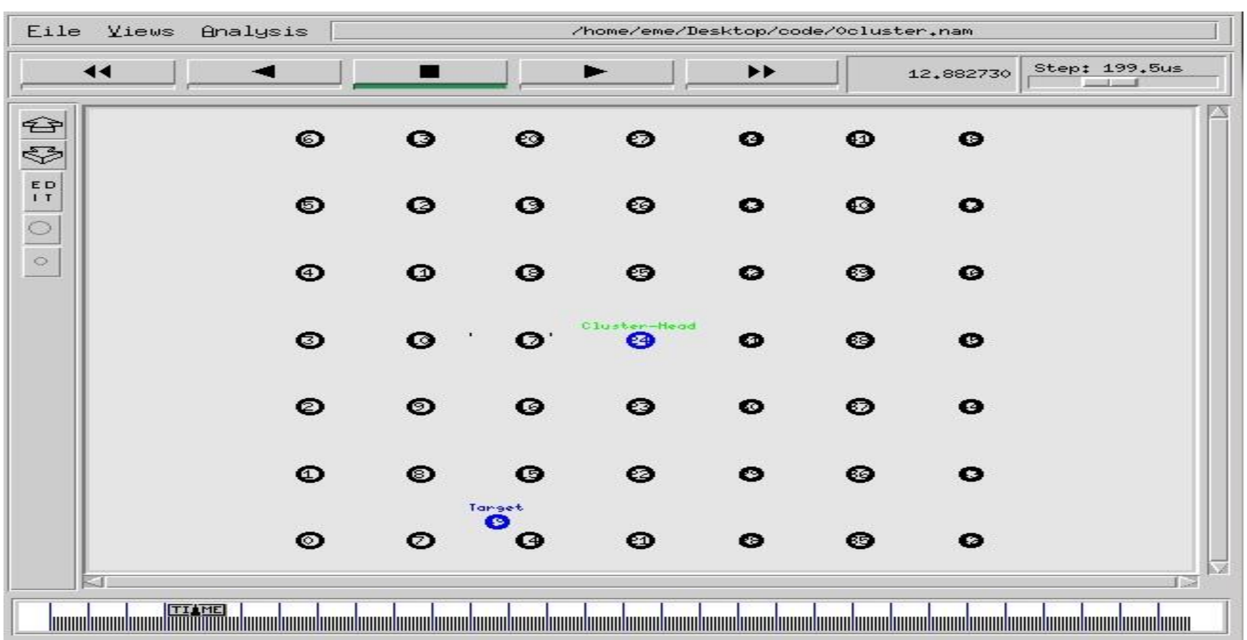


Figure 18: Animation of No-clustering Topology

To run the simulation an automated shell script is given in appendix. After running the simulation resulting NAM for no-cluster, 2-clusters and 4-clusters are shown in figure 18, 19, 20 Respectively.

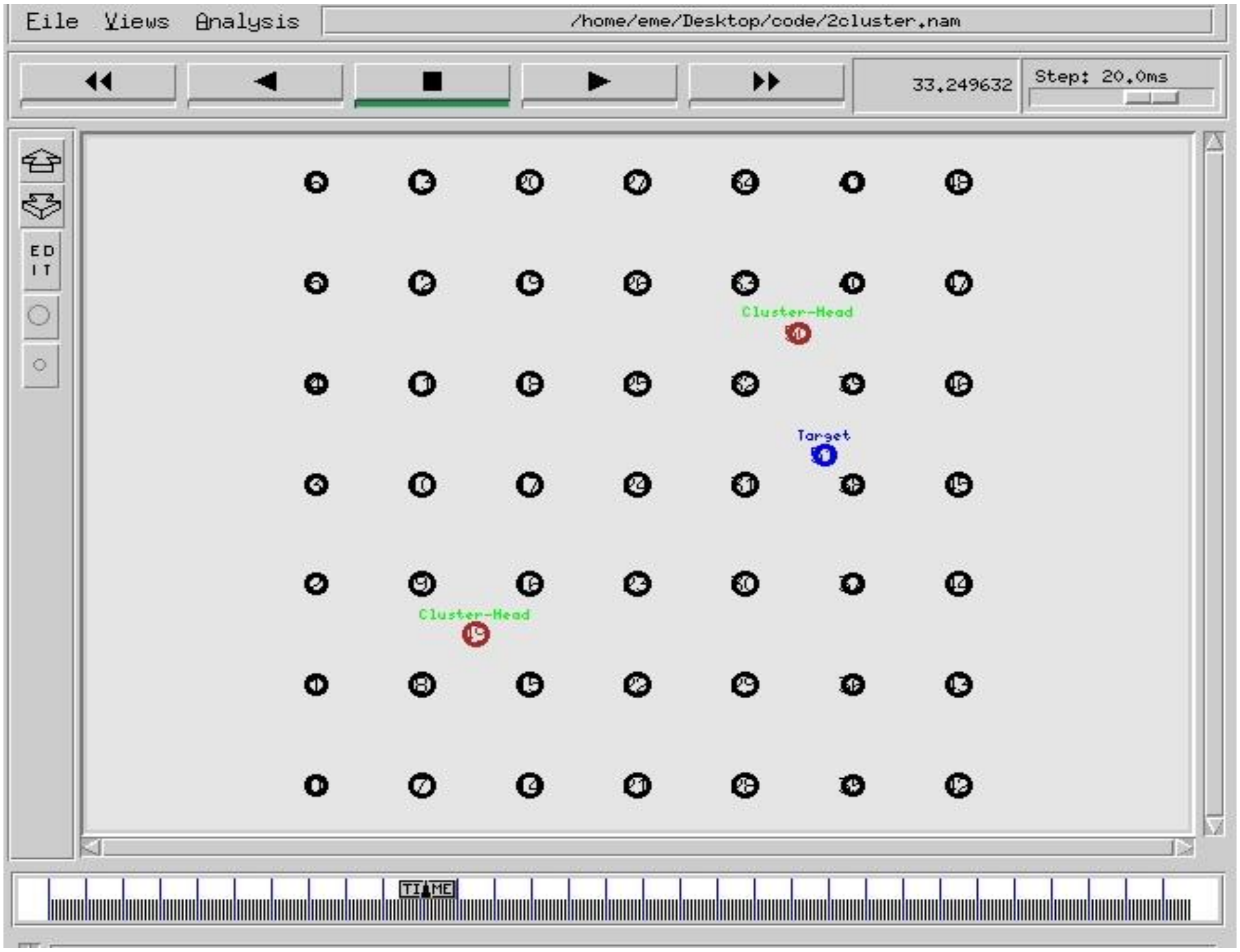


Figure 19: Animation of Two-clusters Topology

In figure 18, node 24 represents the cluster head and node 49 represents the target which is randomly moving in the area. Whereas in figure 19, the topology is divided into two clusters, node 49 and node 50 represents the cluster heads of both clusters and node 51 is the randomly moving target. In figure 20, the entire area is divided into four clusters and node 49, 50, 51 and 52 are the cluster heads. The clusters are defined by their geographic locations, if target lies in the cluster 1 and a node sense the target it will report to its respective cluster-head.

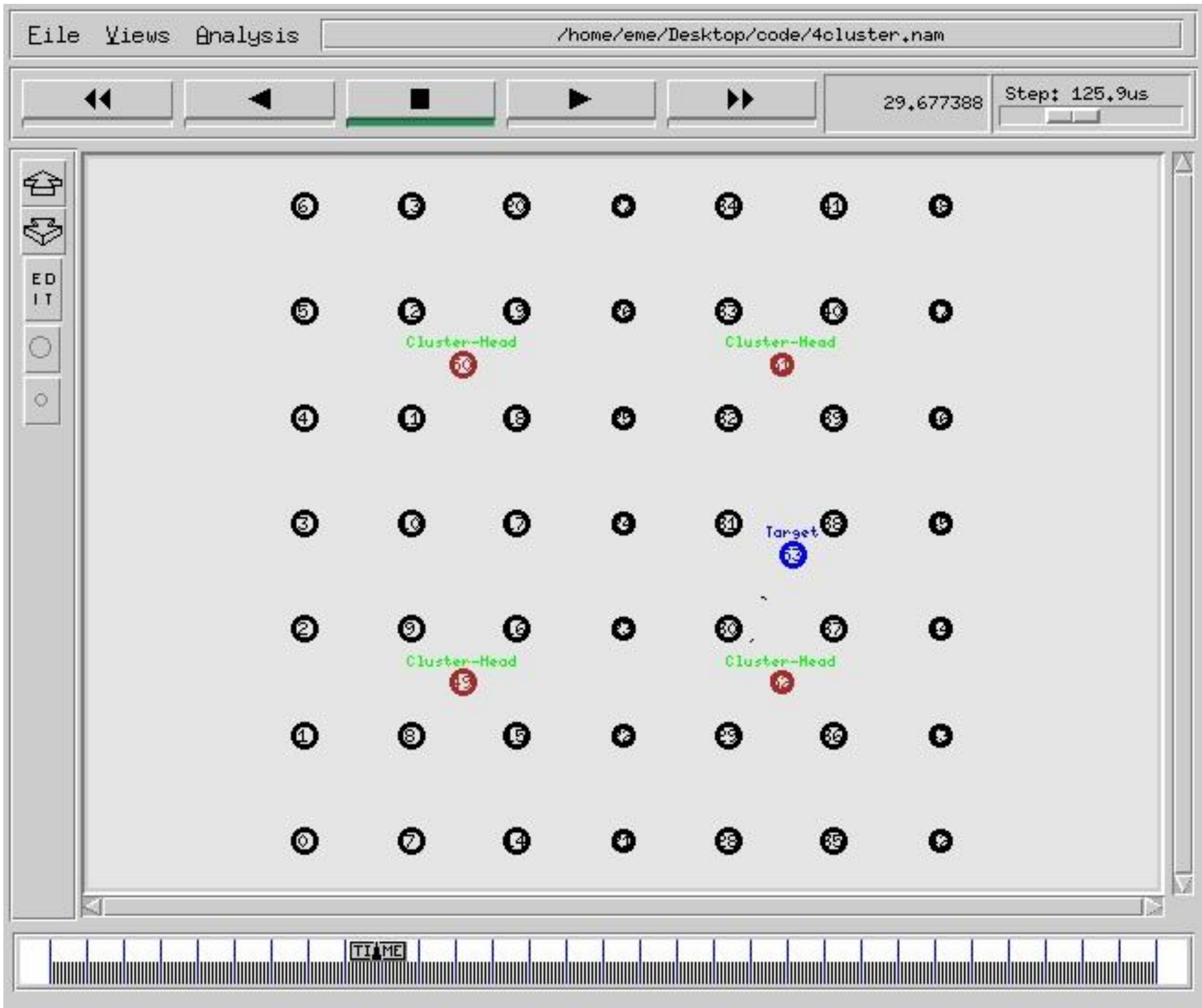


Figure 20: Animation of Four-clusters Topology

CHAPTER 6: RESULTS and FUTURE WORK

To evaluate the performance of WSN, we used the ns2 tool for simulation purposes and for the sake of improvement of results and justify the proposed methodology we divided the WSN topology into geo-graphical based clustering and noted some performance parameters like routing overhead, network life and tracking accuracy.

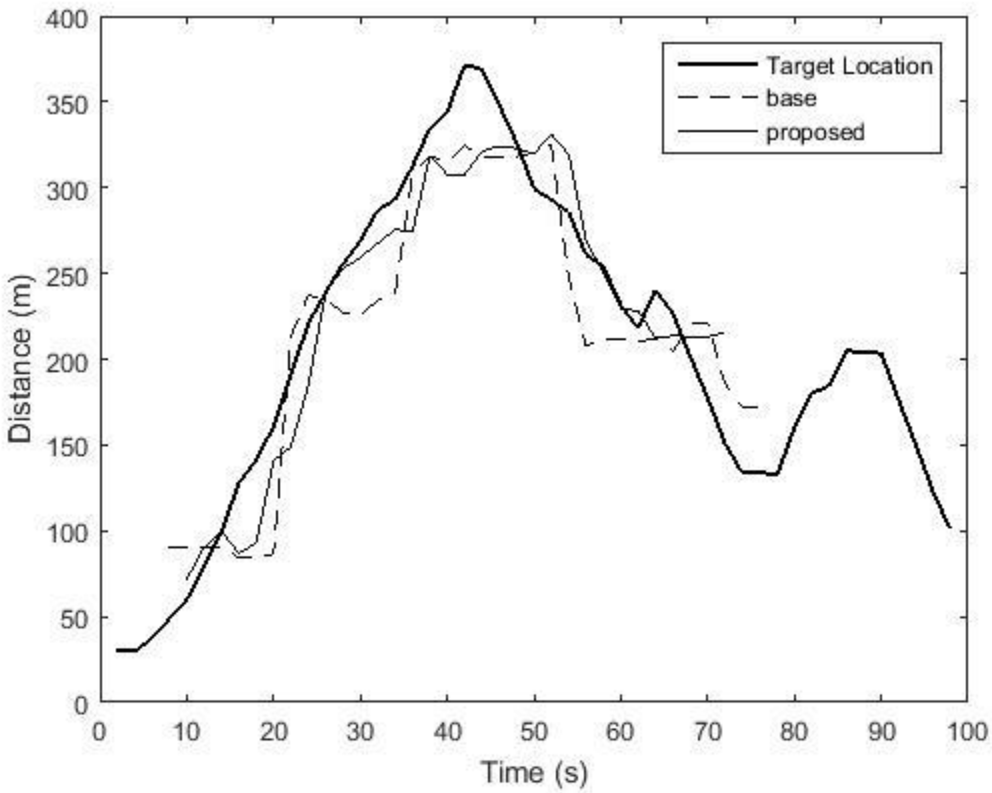


Figure 21: Target Location

Figure 21 illustrates the exact target location and calculated results of base and proposed solution. Tracking error can be keenly observed in the figure 22, initially the error is much higher because of some basic assumption, as long as algorithm works and it received the information from the sensor nodes and based on that information it predicts the movement direction of target and calculates the coordinates of the target, tracking error significantly reduces. Moreover, average error can be observed in the table 3 given below.

Table 3: Average Distance Error

Topology	Avg. Distance Error (%)
Base	7.468
Proposed	5.189

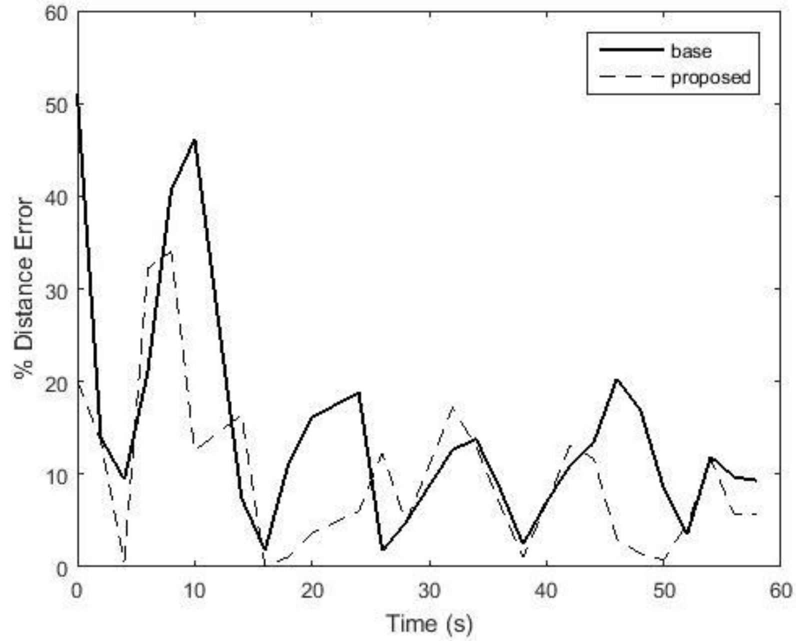


Figure 22: Distance Error

It has been noted that network routing overhead dramatically reduced by 40% after dividing the network into clusters and the simplicity of the algorithm has increased the network life by more than 8% as it can be shown in the figures 23, 24 respectively.

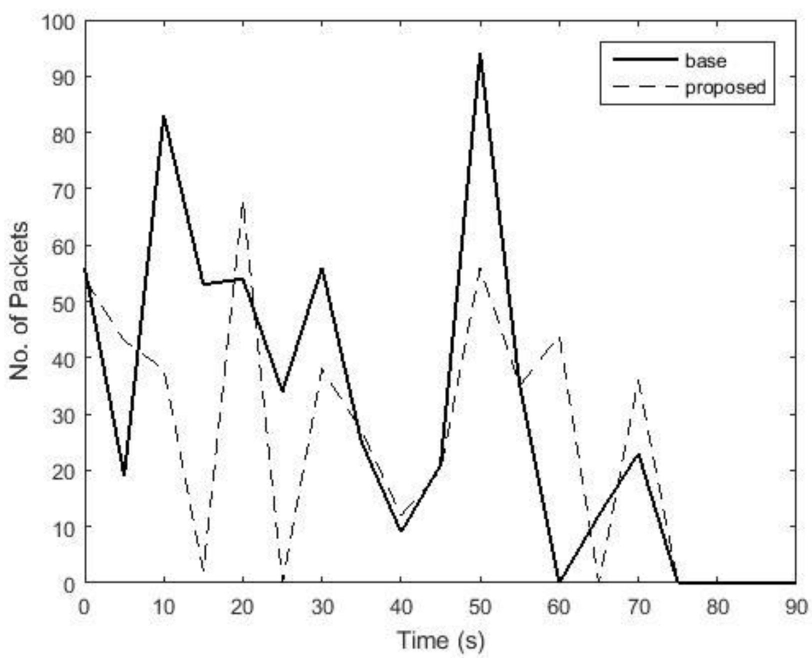


Figure 23: Routing Overhead

Initially node sense the target and calculates the distance based on the received power by using attenuation model. After sensing and distance calculating operation sends the precise information to the cluster-head. It is being observed that during communication consumption of energy resources are extremely high, that's the motive behind this proposed technique by cutting down the overheads and resources can be saved which directly effects the network life of wireless sensor network and can be seen in the figure 24.

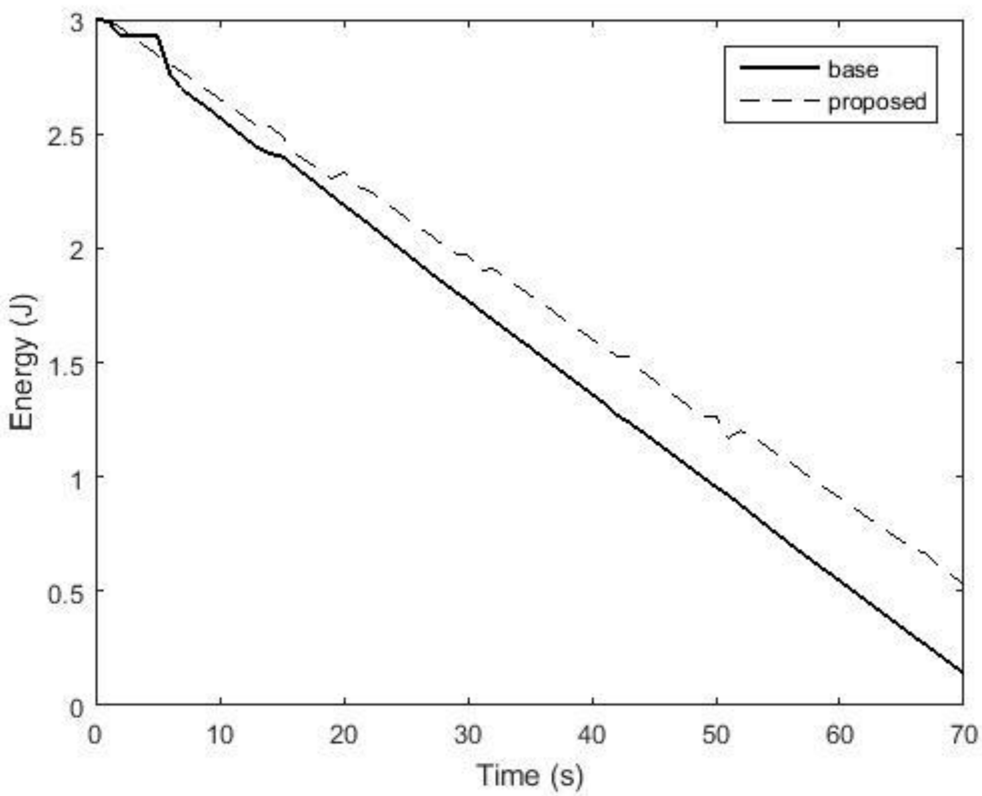


Figure 24: Network Life of WSN

As most of the processing during sensing is distributed and network is divided into clusters. So, the end to end delay during communication reduces as shown in figure 25. Table 4 compares the network performance parameters.

Table 4: Simulation Results

Topology	Network life-time (sec)	Normalized Routing Overhead	E-E Delay (ms)
Base	73.09	1.31	35.29
Proposed	74.85	1.1	11.81

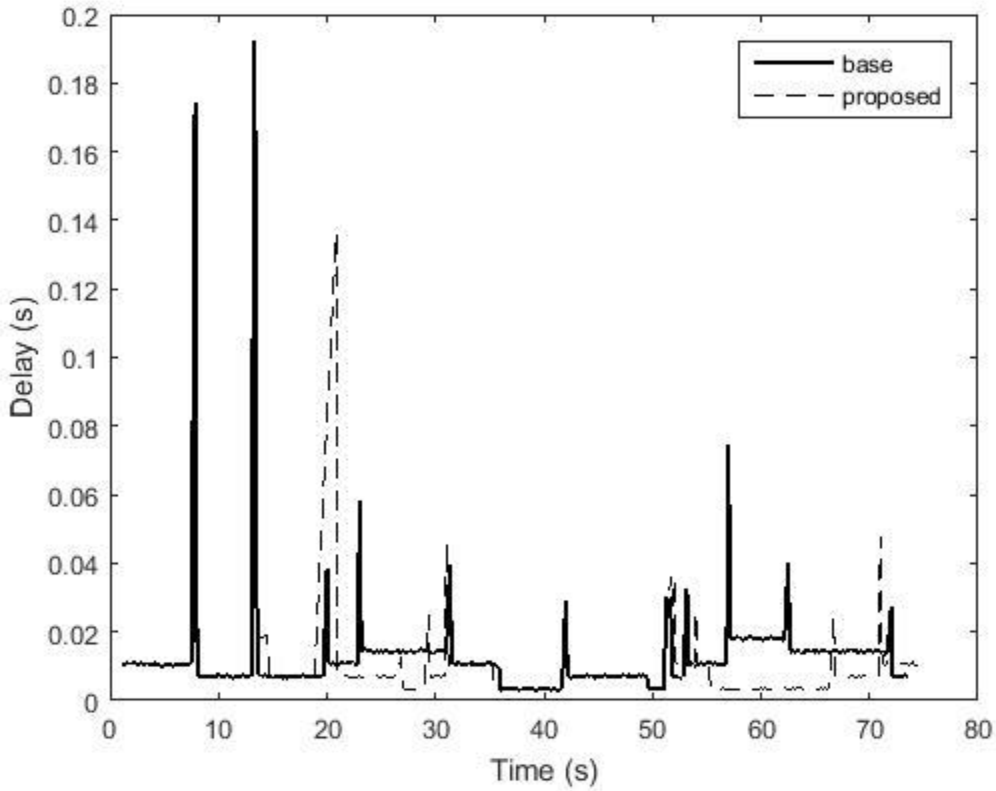


Figure 25: End-to-End Delay

Future work can be extended by working on the different distribution patterns and error can be further reduced and the feasibility of this algorithm with other protocols.

APPENDIX A

7.1 TCL Script

```
#====================================================================
# Define options for WSN Topology
#====================================================================

    set val(chan)          Channel/WirelessChannel     ; #Channel type
    set val(prop)          Propagation/FreeSpace       ; #Radio-propagation model
    set val(ant)           Antenna/OmniAntenna         ; #Antenna type
    set val(ll)            LL                          ; #Link layer type
    set val(ifq)           Queue/DropTail/PriQueue     ; #Interface queue type
    set val(ifqlen)        100                         ; #Max packet in ifq
    set val(netif)         Phy/WirelessPhy             ; #Network interface type
    set val(mac)           Mac/Csma/Ca                 ; #MAC type
    set val(rp)            AODV                        ; #Ad-hoc routing protocol
    set val(nn)            54                          ; #Number of Mobile Nodes
    set val(x)             300                         ; #X dimension to topography
    set val(y)             300                         ; #Y dimension to topography
    set val(stop)          100                         ; #Time of Simulation (Seconds)
    set energymodel        EnergyModel                 ; #Energy Model
    set initialenergy      3.0                         ; #Initial energy in Joules
    set p_rx               0.175                       ; #Receiving Power in Watts
    set p_tx               0.175                       ; #Transmitting Power in Watts
    set i_pw               0.035                       ; #Idle Power
    set s_pw               0.00000175                  ; #Sensing Power

#====================================================================
# Antenna Related Values
#====================================================================
# It works like the 914MHz Lucent WaveLAN DSSS radio interface

    set RxT_               3.652e-10           ; #Receiving Threshold (Hardware feature)
    set Frequency_         914e+6              ; #Signal Frequency (Hardware feature)
    set PI                 3.14                ; #Value of Pi
    set opt(AnH)           1.5                 ; # NS2 by-default Antenna Height
    set opt(Pt)            100.0               ; # Transmission Power/Range in meters
    Phy/WirelessPhy set CPThresh_ 10.0         ; # Capture Threshold
    Phy/WirelessPhy set CSThresh_ 1.8e-9       ; # Carrier Sense Threshold / Sensing range
    Phy/WirelessPhy set RXThresh_ $RxT_        ; # Receiving Threshold
    Phy/WirelessPhy set Rb_ 2*1e6              ; # Bandwidth
    Phy/WirelessPhy set freq_ $Frequency_      ; # Frequency
```

```
        Phy/WirelessPhy set L_ 1.0                         ; # Antenna Loss


        if { $val(prop) == "Propagation/FreeSpace" } {
                set SL_ 300000000.0
                set lambda [expr $SL_/$Frequency_]
                set lambda_2 [expr $lambda*$lambda]
                set CoD_ [expr 4.0*$PI*$opt(AnH)*$opt(AnH)/$lambda]

        if { $opt(Pt) <= $CoD_ } {
                set temp [expr 4.0*$PI*$opt(Pt)]
                set TP_ [expr $RxT_*$temp*$temp/$lambda_2]
                Phy/WirelessPhy set Pt_ $TP_
          } else {
                set d4 [expr $opt(Pt)*$opt(Pt)*$opt(Pt)*$opt(Pt)]
                set hr2ht2 [expr $opt(AnH)*$opt(AnH)*$opt(AnH)*$opt(AnH)]
                set TP_  [expr $d4*$RxT_/$hr2ht2]
                Phy/WirelessPhy set Pt_ $TP_
           }
}
#===================================================================
# Event scheduler object creation
#===================================================================

        set ns [new Simulator]

#===================================================================
# Creating Trace file and NAM file
#===================================================================

        set tracefd [open 4cluster.tr w]
        set namtrace [open 4cluster.nam w]
        $ns trace-all $tracefd
        $ns namtrace-all-wireless $namtrace $val(x) $val(y)

#===================================================================
# Set up topography object
#===================================================================

        set topo [new Topography]
        $topo load_flatgrid $val(x) $val(y)
        set god_ [create-god $val(nn)]

#===================================================================
# Nodes Configuration
#===================================================================
```

```
    $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) \
            -channelType $val(chan) \
            -topoInstance $topo \
            -energyModel  $energymodel \
            -initialEnergy $initialenergy \
            -rxPower $p_rx \
            -txPower $p_tx \
            -idlePower $i_pw \
            -sensePower $s_pw \
            -agentTrace ON \
            -routerTrace ON \
            -macTrace OFF \
            -movementTrace ON

#===================================================================
# Creating node objects
#===================================================================

     for {set i 0} {$i < $val(nn) } { incr i } {
            set node_($i) [$ns node]
     }
     for {set i 0} {$i < [expr $val(nn)-5]  } {incr i } {
            $node_($i) color black
            $ns at 0.0 "$node_($i) color black"
     }
     for {set i 49} {$i < [expr $val(nn)-1]  } {incr i } {
            $node_($i) color brown
            $ns at 0.0 "$node_($i) color brown"
            $ns at 0.0 "$node_($i) label Cluster-Head"
     }
#===================================================================
# Provide initial location of mobile nodes
#===================================================================

     $node_(0) set X_ 0.0
     $node_(0) set Y_ 0.0
     $node_(0) set Z_ 0.0
```

```
	$node_(1) set X_ 0.0
	$node_(1) set Y_ 50.0
	$node_(1) set Z_ 0.0
	.
	.
	.

	$node_(52) set X_ 225.0
	$node_(52) set Y_ 75.0
	$node_(52) set Z_ 0.0

#=====================================================================
# Target Node Random Movement
#=====================================================================

	$node_(53) color blue
	$ns at 0.0 "$node_(53) color blue"
	$ns at 0.0 "$node_(53) label Target"
	$ns at 0.0 "$node_(53) label-color blue"

#Controlled random-way point model
	array set xaxis {0 5 1 8 2 12 3 19 4 23 5 30 6 41 7 48 8 53 9 59 10 63 11 78 12 92 13 99
14 118 15 127 16 131 17 138 18 140 19 163 20 181 21 212 22 223 23 261 24 230 25 223 26 270
27 225 28 228 29 231 30 233 31 270 32 225 33 228 34 231 35 238 36 240 37 242 38 244 39 250
40 260 41 270 42 275 43 263 44 254 45 241 46 225 47 201 48 190 49 181 50 162 51 151 52 143
53 135 54 123 55 105 56 95 57 78 58 71 59 55 60 57 61 51 62 45 63 41 64 31 65 26 66 18 67 17
68 20 69 26 70 29 71 36 72 43 73 21 74 43 75 51 76 63 77 78 78 81 79 94 80 101 81 112 82 120
83 131 84 141 85 147 86 149 87 156 88 161 89 165 90 151 91 43 92 23 93 40 94 53 95 15 96 23
97 71 98 21 99 29 100 11}
	array set yaxis {0 40 1 53 2 15 3 23 4 71 5 21 6 29 7 11 8 23 9 10 10 41 11 24 12 5 13 8
14 12 15 19 16 23 17 30 18 41 19 48 20 53 21 59 22 63 23 78 24 92 25 99 26 118 27 128 28 131
29 138 30 140 31 163 32 181 33 212 34 223 35 227 36 229 37 231 38 223 39 239 40 245 41 255
42 262 43 259 44 257 45 251 46 254 47 241 48 220 49 231 50 242 51 250 52 238 53 253 54 245
55 231 56 260 57 214 58 220 59 215 60 190 61 212 62 227 63 237 64 231 65 225 66 214 67 201
68 190 69 172 70 159 71 141 72 123 73 131 74 140 75 120 76 102 77 112 78 121 79 131 80 137
81 141 82 146 83 128 84 138 85 143 86 138 87 131 88 126 89 119 90 100 91 97 92 92 93 81 94
77 95 63 96 80 97 63 98 59 99 43 100 23}



	for {set i 0} {$i <= $val(stop)} {set i [expr $i+1]} { #target position will changed after 1
seconds
		$ns at $i "$node_(53) setdest $xaxis($i) $yaxis($i) 15.0"
	}

#=====================================================================
```

Communication Between Nodes
#===

```
        for {set i 0} {$i <= $val(stop)} {set i [expr $i+1]} {
        if {($xaxis($i) < 150) && ($yaxis($i) < 150)} {
        set udp1 [$ns create-connection UDP $node_(53) LossMonitor $node_(49) 0]

        $udp1 set fid_ 40
        set cbr1 [$udp1 attach-app Traffic/CBR]
        $cbr1 set rate_ 10kb
        $ns at $i "$cbr1 start"
        $ns at [expr $i+1] "$cbr1 stop"
        }

        if {($xaxis($i) < 150) && ($yaxis($i) > 150)} {
        set udp1 [$ns create-connection UDP $node_(53) LossMonitor $node_(50) 0]

        $udp1 set fid_ 40
        set cbr1 [$udp1 attach-app Traffic/CBR]
        $cbr1 set rate_ 10kb
        $ns at $i "$cbr1 start"
        $ns at [expr $i+1] "$cbr1 stop"
        }

        if {($xaxis($i) > 150) && ($yaxis($i) > 150)} {
        set udp1 [$ns create-connection UDP $node_(53) LossMonitor $node_(51) 0]

        $udp1 set fid_ 40
        set cbr1 [$udp1 attach-app Traffic/CBR]
        $cbr1 set rate_ 10kb                    #sending rate
        $ns at $i "$cbr1 start"
        $ns at [expr $i+1] "$cbr1 stop"
        }
        if {$xaxis($i) > 150 && $yaxis($i) < 150} {
        set udp1 [$ns create-connection UDP $node_(53) LossMonitor $node_(52) 0]

        $udp1 set fid_ 40
        set cbr1 [$udp1 attach-app Traffic/CBR]
        $cbr1 set rate_ 10kb                    #sending rate
        $ns at $i "$cbr1 start"
        $ns at [expr $i+1] "$cbr1 stop"
        }
        }
```

#===
Define node initial position in nam
#===

```
	for {set i 0} {$i < $val(nn)} { incr i } {
		$ns initial_node_pos $node_($i) 10
		}

#===================================================================
# Telling nodes when the simulation ends
#===================================================================

	for {set i 0} {$i < $val(nn) } { incr i } {
		$ns at $val(stop) "$node_($i) reset";
		}

#===================================================================
# Ending NAM and The Simulation
#===================================================================

	$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
	$ns at $val(stop) "stop"
	$ns at 100.1 "puts \"end simulation\"; $ns halt"

#===================================================================
# Stop Procedure
#===================================================================

	proc stop {} {
		global ns tracefd namtrace
		$ns flush-trace
		close $tracefd
		close $namtrace
		#Call xgraph to display the results
		exec awk -f ./awk/location.awk 4cluster.nam >
		./awk/location/data/TargetlocationinCluster4.txt
		exec awk -f ./awk/networkoverhead.awk 4cluster.tr > 4-clusters
		exec awk -f ./awk/4networklife.awk 4cluster.tr > 4-cluster-network-life
		exec nam 4cluster.nam &
		}

#===================================================================
# Run Simulation
#===================================================================

	$ns run
```

7.2 AWK Scripts

7.2.1 Target Location

Target exact location:

```
BEGIN {
            time=2.00000;
}
{

            for (i = time; i < 100; i = i + time) {
                        if ($1=="n" && $3==i)
                        print $3, $7, $9;
                        else continue;
                                                                        }

}
END {
}
```

Trace files alignment:

```
1       NR>3 && $3<48 {print substr($1,1,length($1)-3) , substr($3, 1, length($3)-1)}
2       $1=="r" && $7=="cbr" && $29=="1" {print substr($2,1,length($2)-5), substr($27,1,length($27)-1)}
3       BEGIN{
        }
        {
            time[$3]=$3;
            n[$3]=$4;
            for(i in time){
            if(time[i]==$1){
            s[i]=$2;
            }
            }
        }
        END{
            for(i in time){
                        print i, s[i], n[i];
            }
        }
4       $3>1 {print $0}
5       $2>1 {print $2, $3}
6       {print $2, $3}
```

Target location Algorithm:

```
BEGIN {

#Node_0 Information
            up[0]=1
            down[0]=0
            right[0]=7
            left[0]=0
```

```
        ul[0]=o
        ur[0]=8
        ll[0]=0
        lr[0]=0
        x[0]=0
        y[0]=0

#Node_1 Information
        up[1]=2
        down[1]=0
        right[1]=8
        left[1]=1
        ul[1]=1
        ur[1]=9
        ll[1]=1
        lr[1]=7
        x[1]=0
        y[1]=50
.
.
.

#Node_52 Information
        up[52]=52
        down[52]=52
        right[52]=52
        left[52]=52
        ul[52]=30
        ur[52]=37
        ll[52]=29
        lr[52]=36
        x[52]=225
        y[52]=75
#other parameters
        node=0 #current Node
        time=0
        pnode=0 #previous Node
        nnode=0 #next Node
        stop=100
}

{
        t[$1]=$1; #Time Array
        d[$1]=$2; #Distance Array
        n[$1]=$3; #Node Array
        for(i in n){
        node=n[i];
        time=t[i];
        d1=d[i];
        distance=sqrt((d1^2)/2)
        if(pnode==ll[node]){                # Case No. 1
        nx[time]=x[node]-distance;
        ny[time]=y[node]-distance;
        }
        else if(pnode==down[node]){         # Case No. 2
        nx[time]=x[node];
```

```
            ny[time]=y[node]-distance;
            }
            else if(pnode==lr[node]){               # Case No. 3
            nx[time]=x[node]+distance;
            ny[time]=y[node]-distance;
            }
            else if(pnode==right[node]){            # Case No. 4
            nx[time]=x[node]+distance;
            ny[time]=y[node];
            }
            else if(pnode==ur[node]){               # Case No. 5
            nx[time]=x[node]+distance;
            ny[time]=y[node]+distance;
            }
            else if(pnode==up[node]){               # Case No. 6
            nx[time]=x[node];
            ny[time]=y[node]+distance;
            }
            else if(pnode==ul[node]){               # Case No. 7
            nx[time]=x[node]-distance;
            ny[time]=y[node]+distance;
            }
            else if(pnode==left[node]){             # Case No. 8
            nx[time]=x[node]-distance;
            ny[time]=y[node];
            }
            else if(pnode!=node){
            pnode=node;
            }
}
}

END{
            for(i in n){
            print t[i], nx[i], ny[i];
            }
}
```

7.2.2 Routing Overhead

```
BEGIN{
            count=0;
            start=0;
            a=1; #interval for calculation
}
{
            if(($1 == "s" || $1 == "f") && $4 == "RTR" && $7 =="AODV" && $2<=start+a){
                        count++
                        }
                        else if($2>start+a){
                        count1=count;
                         count=0;
                        start1=start;
                         start=start+a;
                         print start1, count1;
                        }
```

```
}
```

7.2.3 Network Life

1 BEGIN{

```
}
{
    if($7 == "cbr" && $1 == "s" || $1 == "r"){
            t=$2;
            e= $14*1;
            print t, e;
            }
            }
    END{
}
```

2 BEGIN{

```
    i=1
    for(k=1; k<=80; k++){
            val[k]=0;
    }
    }
    {
            t[$1]=$1
            value[$1]=$2

    for(i=1; i<=80; i++){
            if($1<=i && $1>=(i-1)){
                    if(value[$1]>val[i]){
                            val[i]=value[$1]
                                }

                              }

            }
    }
    END{
            for(i in val){
            print i, val[i]
            }
    }
```

7.2.4 End-to-End Delay

```
    BEGIN {
            hpi = 0;
            }
    {
            a = $1;
            t = $2;
            pi = $6;
            if ( pi > hpi ) hpi = pi;
            if ( st[pi] == 0 )  st[pi] = t;
```

```
                    if ( a != "d" ) {
                    if ( a == "r" ) {
                               et[pi] = t;
                     }
          } else {
                              et[pi] = -1;
          }
          }
          END {
          for ( pi = 0; pi <= hpi; pi++ ) {
          start-time = st[pi];
          end-time = et[pi];
          duration = end-time – start-time;
          if ( start-time < end-time ) printf("%f %f\n", start-time, duration);

          }
          }
```

7.2.5 Jitter

```
BEGIN {
      received=0
  }
 {
         e = $1
         t = $2
         id = $6
         size = $8
         type = $4
 if (type == "AGT" && sendTime[id] == 0 && (e == "+" || e == "s") && size >= 200) {
     sendTime[id] = t
 }
 if (type == "AGT" && e == "r" && size >= 200) {
         recvTime[id] = t
         received++
     }
 }

 END {
     jitter1 = jitter2 = tmp_recv = 0
     prev_time = delay = prev_delay = processed = 0
     prev_delay = -1
     for (i=0; processed<received; i++) {
         if(recvTime[i] != 0) {
              tmp_recv++
             if(prev_time != 0) {
                 delay = recvTime[i] - prev_time
                 e2eDelay = recvTime[i] - sendTime[i]
                 if(delay < 0) delay = 0
                 if(prev_delay != -1) {
                 jitter1 += abs(e2eDelay - prev_e2eDelay)
                 jitter2 += abs(delay-prev_delay)
                 }
                 prev_delay = delay
                 prev_e2eDelay = e2eDelay
```

```
            }
            prev_time = recvTime[i]
        }
        processed++
    }
}

END {

    printf("Mean Jitter(ms) = %.2f\n",jitter1*1000/tmp_recv);
    printf("Current Jitter(ms) = %.2f\n",jitter2*1000/tmp_recv);
}

function abs(value) {
    if (value < 0) value = 0-value
    return value
}
```

7.2.6 Throughput

```
BEGIN {
    size = 0
    start = 1e6
    stop = 0
}
{
        e = $1
        t = $2
        node_id = $3
        pkt_id = $6
        psize = $8
        type = $4

if (type == "AGT" && (e == "+" || e == "s") && psize >= 210) {
  if (t < start) {
        start = t
        }
    }

if (type == "AGT" && e == "r" && psize >= 210) {
    if (t > stop) {
        stop = t
        }
    hsize = psize % 210
    psize -= hsize
    size += psize
    }
}

END {
    printf("Average Throughput[kbps] = %.5f\t\t Start=%.5f\tStop=%.5f\n",( size /(stop-start))*(8/1000),start,stop)
}
```

REFERENCES

[1] Ahmadi, E., Sabaei, M., Ahmadi, M. H., May 2011, "A New Adaptive Method for Target Tracking in Wireless Sensor Networks," International Journal of Computer Applications (0975 – 8887), vol. 22, No. 9

[2] Chen, M. X., Wang, Y. D., 2009, "An Efficient location tracking structure for wireless sensor networks," Computer Communications vol. 32 p.1495-1504

[3] Nguyen, L. T., Defago, X., Beuran, R., Shinoda, Y., 2008, "An Energy Efficient Routing Scheme for Mobile Wireless Sensor Networks," IEEE International Symposium on Wireless Communication Systems.

[4] Yu, Z., Wei, J., Liu, H., 2009, "An Energy-Efficient Target Tracking Framework in Wireless Sensor Networks," EURASIP Journal on Advances in Signal Processing

[5] Zhang, W., Cao, G., September 2004, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," IEEE Transactions on Wireless Communications vol. 3, No. 5

[6] Lee, J., Cho, K., Lee, S., Kwon, T., Choi, Y., August 2006, "Distributed and energy-efficient target localization and tracking in wireless sensor networks," Computer Communications vol. 29, pages 2494-2505

[7] Bhuiyan, M. Z. A., Wang, G., Wu, J., August 2009, "Target Tracking with Monitor and Backup Sensors in Wireless Sensor Networks," Proceedings of 18th International Conference on Computer Communications and Networks

[8] Khera, S., Mehla, Dr. N., Kaur, Dr. N., June 2016, "Applications and Challenges in Wireless Sensor Networks," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5

[9] Ahmed, M. R., Huang, X., Sharma, D., Cui, H., 2012, "Wireless Sensor Network: Characteristics and Architectures," International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, vol. 6, No. 12

[10] X. Huang, M. Ahmed, D. Sharma, "Timing Control for Protecting from Internal Attacks in Wireless Sensor Networks", IEE, ICOIN 2012, Bali, Indonesia, February 2012.

[11] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E, Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, August 2002.

[12] T. He, J. A. Stankovic, C. Lu, T. Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks". IN: Proc. Of 23rd Int'l Conf. on Distributed Computing Systems. Rhode Island: IEEE Computer Society, 2003.

[13] "The Network Simulator – ns- 2", Retrieved July 10, 2016, from http://www.isi.edu/nsnam/ns/

[14] T. Issariyakul, E. Hossain, "Introduction to Network Simulator".

[15] K. Fall, K. Varadhan, "NS2 Manual".

[16] http://www.lchr.org/a/54/9f/project/software.html

[17] The A-Z of Programming Language: AWK. Retrieved June 18, 2016, from http://www.computerworld.com.au/index.php/id;1726534212;pp;2

[18] Workflow of Awk Programming. Retrieved June 20, 2016, from http://www.tutorialspoint.com/awk/awk_workflow.htm

[19] Xgraph command-line options. Retrieved August 30, 2016, from www.xgraph.org

[20] Heinzelman, W., Chandrakasan, A., and Balakrishnan, H., "Energy-Efficient Communication Protocols for Wireless Microsensor Networks", Proceedings of the 33rd Hawaaian International Conference on Systems Science (HICSS), January 2000.

[21] Wireless Sensor Network. Retrieved July 3, 2016, from http://searchdatacenter.techtarget.com/definition/sensor-network

[22] AWK. Retrieved July 16, 2016, from https://en.wikipedia.org/wiki/AWK

[23] Feng Zhao1 and Jaewon ShinEt al IEEE Signal Processing Magazine, March 2002, "Information-Driven Dynamic Sensor Collaboration for Tracking Applications".

[24] M. Walchli, P. Skoczylas, M. Meer and T. Braun, "Distributed Event Localization and Tracking with Wireless Sensors," in Proceedings of the 5th international Conference on Wired/Wireless internet Communications, May 23 - 25, 2007.

[25] E. Olule, G. Wang, M. Guo and M. Dong, "RARE: An Energy Efficient Target Tracking Protocol for Wireless Sensor Networks," 2007 International Conference on Parallel Processing Workshops (ICPPW 2007), 2007.

[26] Jianxun Li and Yan Zhou (2010). Target Tracking in Wireless Sensor Networks, Wireless Sensor Networks: Application - Centric Design, Yen Kheng Tan (Ed.), InTech, DOI: 10.5772/13701. Available from: http://www.intechopen.com/books/wireless-sensor-networks-application-centric-design/target-tracking-in-wireless-sensor-networks.

[27] T Issariyakul and E Hossain, "Introduction to Network Simulator NS2," Springer, Oct. 2008, ISBN: 978-0-387-71759-3.

[28] NS2 Tutorial by Marc Greis, URL: http://www.isi.edu/nsnam/ns/tutorial/ (retrieved Oct 19, 2013).

[29] F. Aghaeipoor, M. Mohammadi, V. Sattari Naeini, “Target tracking in noisy wireless sensor network using artificial neural network”. IEEE, 7th International Symposium on Telecommunications (IST), Sep 2014.