JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A FUZZY LOGIC-BASED ON-DEMAND CHARGING ALGORITHM FOR WIRELESS RECHARGEABLE SENSOR NETWORKS WITH MULTIPLE CHARGERS

M. KAMARUNISHA MCA.,M.phil.,(Ph.D)[1],ANU KARTHIKA R[2],

1 ASSISTANT PROFESSOR, 2PG STUDENT,

DEPARTMENT OF COMPUTER APPLICATIONS

DHANALAKSHMI SRINIVASAN COLLOGE OF ARTS AND SCIENCE FOR WOMAN (AUTONOMOUS), PERAMBALUR, TAMILNADU, INDIA

Abstract

Mobile chargers have greatly promoted the wireless rechargeable sensor networks (WRSNs). While most recent works have focused on recharging the WRSNs in an on-demand fashion, little attention has been paid on joint consideration of multiple mobile chargers (MCs) and multi-node energy transfer for determining the charging schedule of energy-hungry nodes. Moreover, most of the schemes leave out the contemplation of multiple networks attributes while making scheduling decisions and even they overlook the issue of ill-timed charging response to the nodes with uneven energy consumption rates. In this paper, we address the aforesaid issues together and propose a novel scheduling scheme for on-demand charging in WRSNs. We first present an efficient network partitioning method for distributing the MCs so as to evenly balance their workload. We next adopt the fuzzy logic which blends various network attributes for determining the charging schedule of the MCs. We also formulate an expression to determine the charging threshold for the nodes that vary depending on their energy consumption rate. Extensive simulations are conducted to demonstrate the effectiveness and competitiveness of our scheme. The comparison results reveal that the proposed scheme improves charging performance compared to the state-of-the-art schemes with respect to various performance metrics.

Keywords: Mobile chargers, Wireless & On - Demand

1. INTRODUCTION

Wireless sensor nodes widely exist in the Internet of things, these nodes are typically battery-powered and the amount of power they can use is relatively limited. Generally, most of the wireless sensor nodes are deployed outdoors, i.e., trees by the river, the roof of buildings and objects in the water. In these conditions, the battery cannot be replaced easily by us because of sensor nodes housed in a sealed waterproof case. As the development of the Internet of things, there are a trillion wireless sensor nodes all around the world while a million nodes need daily battery replacements. It is difficult and unrealistic to change batteries one by one in practical appliance. Therefore, how to prolong the network survival time have become an urgent problem and also a research hot area. Wireless sensor networks (WSNs) are being widely used in environmental monitoring forest fire warning and medical care with the vigorous development of wireless communication technology, sensor technology, and microelectronic technology. However, the energy supply modes for sensor node are restricted, which seriously affects the performance and development of sensor networks. Scholars have conducted considerable research for the sensor network energy problem. These works can be roughly divided into three categories: energy saving methods energy collection methods [6], and wireless charging methods.

Energy saving methods usually sacrifices a certain amount of network performance and the energy of sensor nodes limits the increase in network lifetime. The energy conversion method has low energy conversion efficiency and uncontrollable energy acquisition, and accurate prediction is difficult. The wireless charging method is equipped with an active charging power node in the network, and the static or mobile charging node actively provides the sensor node with efficient and timely charging services. The charging process is controllable and predictable. Recent advances in wireless charging techniques and rechargeable lithium-ion battery technology have produced a new solution for the energy problem of wireless sensor networks, and wireless rechargeable sensor networks (WRSNs) have emerged.

With the rapid development of microelectronics technology, signal processing technology, wireless communication technology and computer networks, wireless sensor networks have emerged. Wireless Sensor Network (WSN) is a multi-hop self-organizing network system formed by wireless communication. The sensor node cooperatively senses, collects, and processes information of the perceived object in the network coverage area and sends it to the observer. Wireless sensor networks are widely used in many important fields such as forest fire detection, animal tracking, military area monitoring, early earthquake detection, and border monitoring due to their low cost, low power consumption and multi-function. But as the size of the network increases, it becomes increasingly difficult and unrealistic to periodically replace batteries for all nodes, and limited battery energy will eventually lead to limited network life. In order to solve the problem of limited life in wireless sensor networks, many scholars at home and abroad have conducted a lot of research, and the solutions can be divided into three categories: node energy saving, natural energy collection and wireless charging. The energy-saving method mainly reduces the energy loss per unit time by compressing the transmitted data packets, clustering the network, and selecting the dynamic cluster head. The natural energy harvesting method requires the use of an energy converter on the node through which it can harvest energy (such as solar energy, wind energy, etc.) from the natural environment to extend its life. However, obtaining energy from the natural environment will bring uncertainty in the energy source (such as day and night, strong winds and weak winds), and the energy conversion efficiency is not high. The wireless charging method refers to providing the network with a wireless charging source (such as a static charging station, a mobile charging car, etc.), the charging source travels in the network according to the charging trajectory, and charges the nodes in the network during the driving process

The current research mainly focuses on the following two aspects: (1) Energy saving. For example, nodes working in a low duty cycle mode maximize energy savings and prolong the lifetime of the network by reducing the power consumption of sensor node. In addition, some methods are scheduled according to the length of the data packet, so as to obtain a reasonable and effective energy management strategy. However, the sensor data cannot be uploaded to the sink node or the cloud server of the Internet of things in real-time. (2) Energy harvesting. Another way to solve the energy problem is powered from multiple low-level energy harvesting sources, including solar energy, wind energy, vibration energy, RF energy and more [4], especially in the usage of wireless power technology in recent years. Currently, the emphasis has been placed on energy harvesting as the most important means of maintaining the wireless sensor network. The Wireless Rechargeable Sensor Network (WRSN) has become a newly emerging research area of sensor network that utilizes RF charging as a convenient and efficient charging method for sensor nodes or devices.

2. SYSTEM ANALYSIS

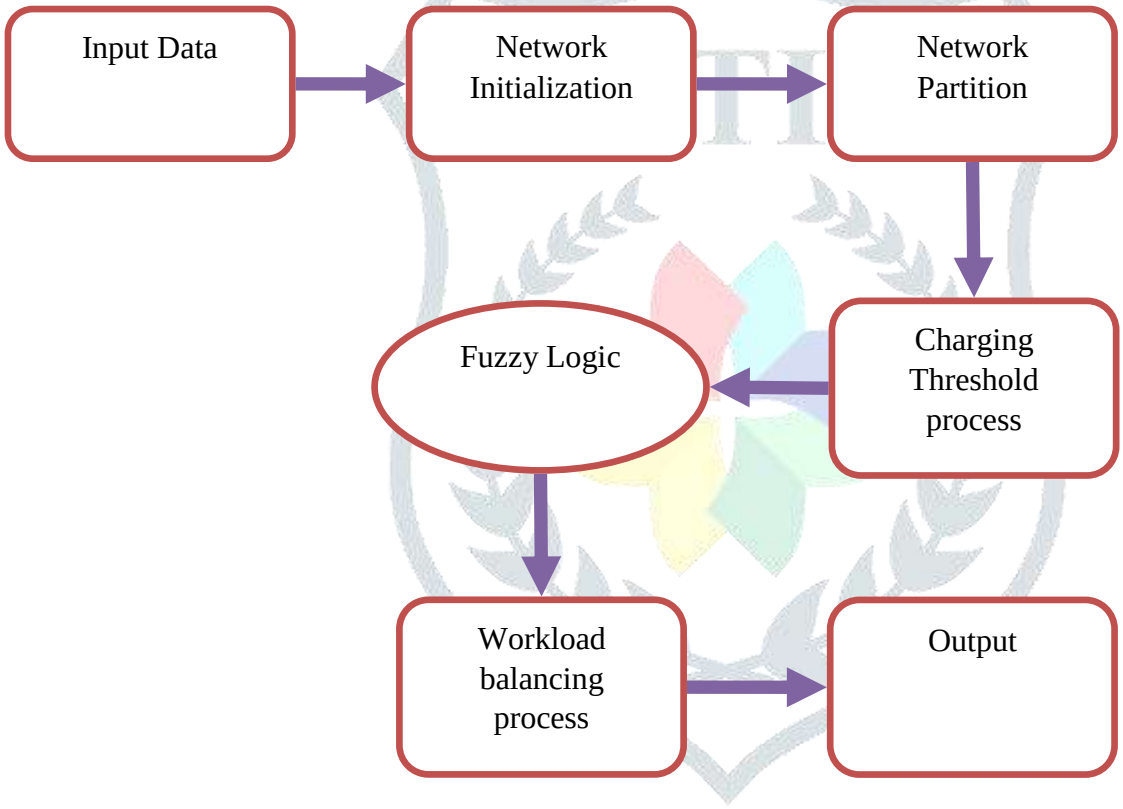
2.1 EXISTING SYSTEM

The MCs in a WRSN usually follow two basic schemes, periodic charging and on-demand charging. In periodic charging, schedule of an MC is known a priori and the MC roams accordingly around the network in a periodic fashion. However, due to heterogeneous and dynamic energy consumption rate of the nodes, the fixed charging schedule is not optimal. This leads to high node failure rate and diminishes the charging performance. On the contrary, an MC in on-demand charging can take real-time charging decision to fulfil energy demand of the sensor nodes.

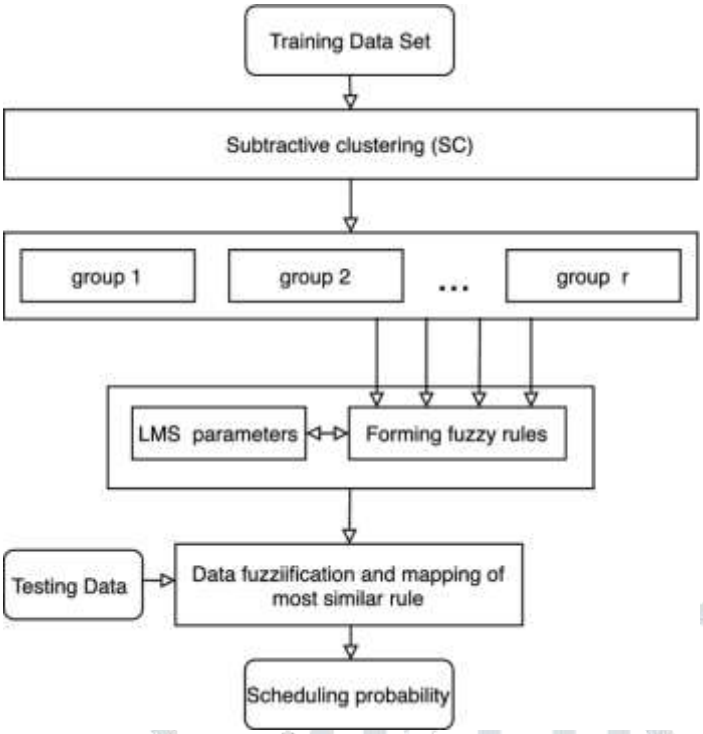
2.2 PROPOSED SYSTEM

A formulation of calculating an adaptive threshold for the charging request is also presented with the aim of timely recharging the nodes with uneven and dynamic energy consumption rates. The proposed scheme is evaluated through extensive simulations and compared with two existing and similar works in terms of energy usage efficiency, average charging latency and survival rate. The results are also statistically validated through analysis of variance (ANOVA) test.

2.3 ARCHITECTURE DIAGRAM



2.4 FLOW DIAGRAM



2.5 PROPOSED PROCESS EXPLANATION

Wireless rechargeable sensor network

In a wireless rechargeable sensor network, it is assumed that there are S chargeable sensors distributed in a two-dimensional area, and the positions of the sensors are known. There are M omnidirectional wireless chargers to be arranged, each of which can be placed anywhere in the area and can be oriented arbitrarily. Assuming that the charging area of each charger covers a subset of the sensors, adjacent chargers can cover a common sensor, which means that a sensor may be simultaneously covered by multiple chargers for charging.

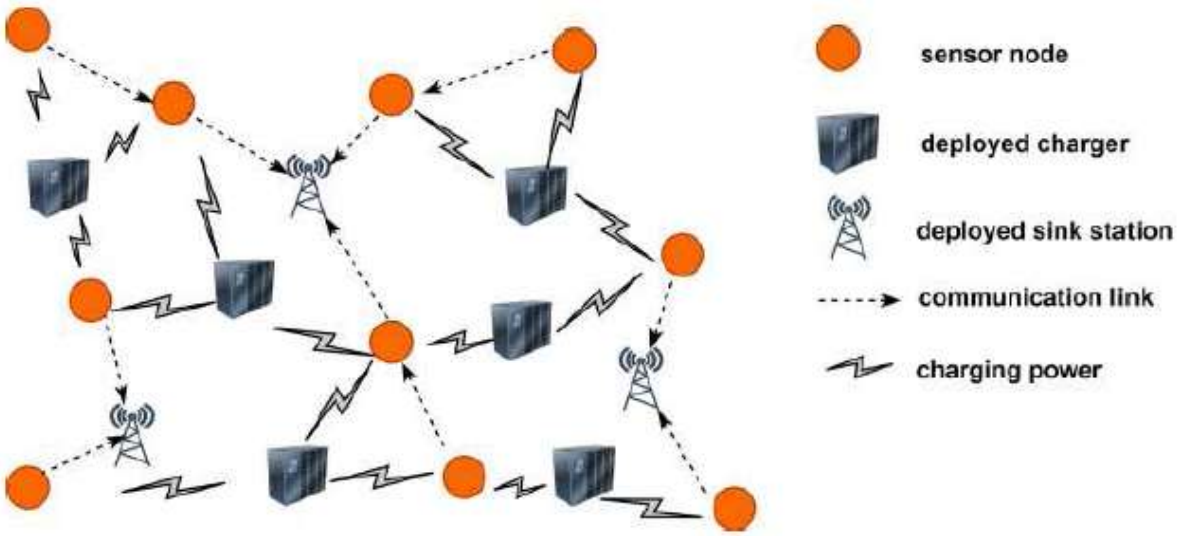


Fig:1 Deployment of wireless rechargeable sensor nodes

Charging Path Allocation

Among many results of SCBMC (Safe Charging Based on Multiple Mobile Chargers) algorithm based on MTSP-fixed, selecting the uniform distribution of sub path as the final path allocation. Considering the energy using the process of moving action is more than the energy using charging consumption for every node. In order to meet the challenge of nodes' continual work, except for the timely charging work, the charging circuit or capacitor should keep the charging power above the threshold of Pmax in the duty cycle. Then when

these chargers will charge for every node, firstly they judge the safety status in the neighbour area. After confirming the safety, the chargeable sensor nodes can request for charging

2.6 PROPOSED SCHEME

The proposed scheme has the following basic operational steps: (1) after network initialization, it first partitions the network to evenly distribute the charging load of sensor nodes among the MCs, (2) it then estimates a dynamic charging threshold for nodes based on their heterogeneous energy consumption rate, and (3) finally, it evaluates various network attributes jointly using fuzzy logic in order to determine most suitable next-to-be-charged node.

2.7 Fuzzy Logic system

Fuzzy Logic is defined as a many-valued logic form which may have truth values of variables in any real number between 0 and 1. It is the handle concept of partial truth. In real life, we may come across a situation where we can't decide whether the statement is true or false. At that time, fuzzy logic offers very valuable flexibility for reasoning.

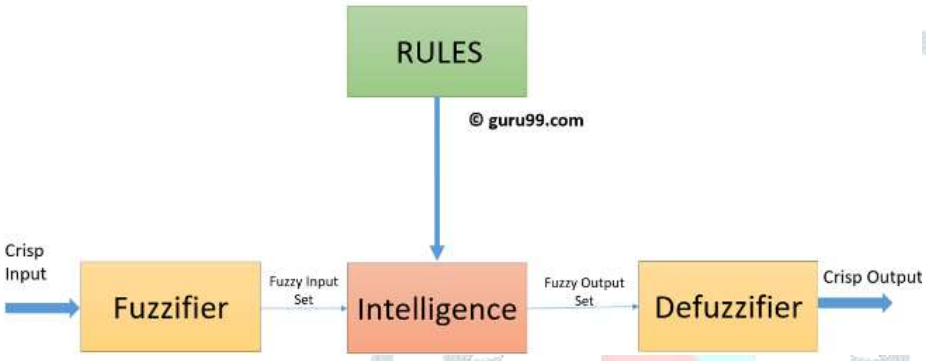


Fig: 2 Fuzzy logic system

Fuzzy logic algorithm helps to solve a problem after considering all available data. Then it takes the best possible decision for the given the input. The FL method imitates the way of decision making in a human which consider all the possibilities between digital values T and F.

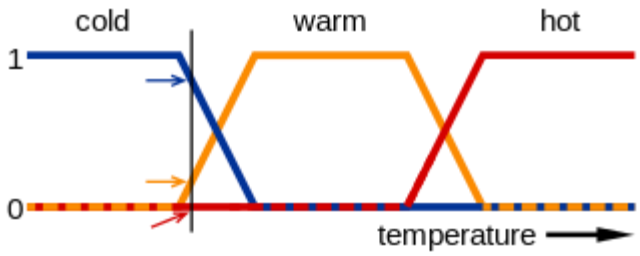
Fuzzy Logic architecture has four main parts as shown in the diagram:

2.8 Fuzzification

Fuzzification step helps to convert inputs. It allows you to convert, crisp numbers into fuzzy sets. Crisp inputs measured by sensors and passed into the control system for further processing. Like Room temperature, pressure, etc.

Fuzzification is the process of assigning the numerical input of a system to fuzzy sets with some degree of membership. This degree of membership may be anywhere within the interval [0,1]. If it is 0 then the value does not belong to the given fuzzy set, and if it is 1 then the value completely belongs within the fuzzy set. Any value between 0 and 1 represents the degree of uncertainty that the value belongs in the set. These fuzzy sets are typically described by words, and so by assigning the system input to fuzzy sets, we can reason with it in a linguistically natural manner.

For example, in the image below the meanings of the expressions cold, warm, and hot are represented by functions mapping a temperature scale. A point on that scale has three "truth values"—one for each of the three functions. The vertical line in the image represents a particular temperature that the three arrows (truth values) gauge. Since the red arrow points to zero, this temperature may be interpreted as "not hot"; i.e. this temperature has zero membership in the fuzzy set "hot". The orange arrow (pointing at 0.2) may describe it as "slightly warm" and the blue arrow (pointing at 0.8) "fairly cold". Therefore, this temperature has 0.2 membership in the fuzzy set "warm" and 0.8 membership in the fuzzy set "cold". The degree of membership assigned for each fuzzy set is the result of fuzzification



3. REQUIREMENTS

HARDWARE REQUIREMENTS

Processor: Dual core processor

Hard disk: 160 GB

RAM: 2 GB

SOFTWARE REQUIREMENTS

Matlab 2013

Windows 7

3.1 SOFTWARE DESCRIPTION

MATLAB

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems. MATLAB is a wonderful environment for serious numerical computations as well as for graphics. It is replacing FORTRAN and other languages that have often been used for numerical scientific computations.

GETTING HELP

MATLAB has several options for on-line assistance. MATLAB offers a tutorial, which can be accessed from the Help menu or by typing ‘demo’ at the command prompt. It would be a good idea to run through some of these demos to get an idea of how MATLAB does “stuff”! The index of MATLAB help information can be accessed from Help -> MATLAB Help. Here you can find information on getting started, using MATLAB, and implementing built-in functions. Information on the built-in functions may also be obtained by using the commands ‘help’ and ‘look for’ in the command window. ‘Help function name’ provides help on the function if you know its exact name. If you don’t know the exact name of the function, use ‘look for keyword’ to get a list of functions with string keyword in their description.

MATLAB PROGRAMS

A MATLAB program, called an M-file, is just a list of MATLAB commands, the same commands that you can use interactively in the command window. You can write that list of commands in a text file and then execute the program from the command window. To create an M-file, start MATLAB and under the File menu select New and M-file. Type your commands in the M-file window. Store the M-file with the same name as the function, and with the suffix ".M" such as "Prog.M". The file can be stored and run from a floppy or the hard disk. After you have changed an M-file, remember to save it before using the function or script. MATLAB uses the saved version of the program, and not the version displayed in the window.

MATLAB programs in M-files can be classified into two groups: **script files** and **function files**. They differ in two things: (i) the way you execute them, and (ii) the type of variables they involve.

SCRIPT FILES

Script files are M-files that can be executed by typing their names in the command window, or calling them from other M-files. The variables they contain or define are global variables. That is, after you execute a script file all variables involved would be in memory and usable from the command window.

3.2 TEST RESULT AND ANALYSIS

TESTING

A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. Achieving an error-free program is the responsibility of the programmer. Program testing checks for two types of errors: syntax and logic.

When a program is tested, the actual output with the expected output is going to compare. When there is discrepancy, the sequence of instructions must be traced to determine the problem. Breaking the program down into self-contained portions, each of which can be checked at certain key points, facilitates the process. The idea is to compare program values against desk-calculated values to isolate the problem.

Testing is an important stage in the system development life cycle (SDLC). The test case is a set of data that a system will process as normal input. As its philosophy behind testing is to find errors the data are created with the express intent of determining whether the system will process them correctly.

Software testing is an important element of software quality assurance and represents the ultimate review of specification, design and loading. The increasing visibility of software AR a system element and the costs associated with a software failure are motivating for well planned through testing.

3.3 TEST OBJECTIVES

These are several rules that can save as testing objectives they are: Testing is a process of executing program with the intent of finding an error. A good test case is one that has a high probability of finding an undiscovered error. If testing is conducted successfully according to the objectives as stated above, it would in cover errors in the software also testing demonstrator that software functions appear to the working according to specification that performance requirements appear to have been met.

3.4 PROGRAM TESTING

There are three ways to test a program

1. for correctness
2. For implementation, efficiency and
3. For Computations complex city.

Test for correctness is supposed to verify that a program does actually what it is designed to do. This is much more difficult than it May appear at first, especially for large programs. Test for implementation efficiency attempt to find ways to make a correct program faster or use less storage.

3.5 TESTING AND CORRECTNESS

The following ideas should be a fact of any testing plan.

Preventive measures

Spot-checks
Testing all parts of the program
Test data
Looking for trouble
Time for testing

The entire testing process can be divided into three phases.

Unit Testing
Integrated Testing
Final/System Testing

3.6 TEST CASES

3.6.1 SYSTEM TESTING

System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

3.6.2 UNIT TESTING

In unit testing, the entire program that makes the system tested. Unit testing first focuses on the modules, independent of one another to locate errors. This enables to detect errors in coding and the logic within the module alone. In the unit testing control path are tested to remove errors within the boundary of the module.

3.6.3 INTEGRATION TESTING

Integration testing can proceed in a number of different ways, which can be broadly characterized as top down or bottom up. On top down integration testing the high level control routines are tested first, possibly with the middle level control structures present only as stubs.

3.6.4 FUNCTIONAL TESTING

Functional testing is a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (Not like in white-box testing).

3.6.5 WHITE BOX TESTING

This is a test case design method that uses the control structure of the procedural design to derive test cases. Using it, the software engineer can derive test cases that, guarantee that all independent paths within a module have been exercised once.

3.6.6 BLACK BOX TESTING

This focuses on the functional requirements of the software. It enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. It attempts to find errors such as:

Incorrect or missing functions
Interface errors
Errors in data structures or external database access

3.7 ANALYSIS

Test analysis is the process of looking at something that can be used to derive test information. This basis for the tests is called the **test basis**.

The test basis is the information we need in order to start the test analysis and create our own test cases. Basically it's a documentation on which test cases are based, such as requirements, design specifications, product risk analysis, architecture and interfaces.

We can use the test basis documents to understand what the system should do once built. The test basis includes whatever the tests are based on. Sometimes tests can be based on experienced user's knowledge of the system which may not be documented.

From testing perspective, we look at the test basis in order to see what could be tested. These are the test conditions. A **test condition** is simply something that we could test.

While identifying the test conditions we want to identify as many conditions as we can and then we select about which one to take forward and combine into test cases. We could call them **test possibilities.**

As we know that testing everything is an impractical goal, which is known as exhaustive testing. We cannot test everything we have to select a subset of all possible tests. In practice the subset we select may be a very small subset and yet it has to have a high probability of finding most of the defects in a system. Hence we need some intelligent thought process to guide our selection called **test techniques.** The test conditions that are chosen will depend on the test strategy or detailed test approach. For example, they might be based on risk, models of the system, etc

3.8 IMPLEMENTATION

Implementation is that stage of the project when the theoretical design is turned into a working system. After testing the modules successfully, the necessary privileges are given to the users. All the users are requested to handle the system carefully. The real time problems that occur are successfully solved. The objective is to put the tested system into operation. It consists of

Testing the developed program with sample data.
Detection and corrections of errors.
Making necessary changes in the system.
Checking of reports.
Creating computer compatible files.
Installation of hardware and software utilities.

An implementation description that shows implementation details for each operation implied by message that is passes to an object. Implementation details include information about the objects private part; that is, internal details about the data structures that describe the objects attributes and procedural details that describe operations.

An implementation description of an object provides the internal details that are required for implementation but are not necessary for invocation. That is, the designer of the object must provide an implementation description and must therefore create the internal details of the object. However, another designer or implementer who uses the object or other instances of the object requires only the protocol description but not the implementation description.

This system is implemented by installing the software on a machine with Windows 2000 environment and connected to a network. The application is run to check if it retrieves the necessary information from remote machines and thereby the application is tested to check for consistency of output and for various kinds of input data.

The module concerning the remote access of the server is also implemented in an internet installed environment. The entire desktop control of any server of any network is retrieved enabling to control the entire network in the Internet. The software is implemented by giving the IP Address of the server foreign countries and found to work as intended. The grid resolution facility and the compression level also working in a very perfect manner.

It is therefore advised to connect the server in the Internet by IP Address and then control the network which enables the desktop contrail of the hosts also. The server can be connected in any system provided with an Internet server.

4. CONCLUSION

Taking into account the MC's limited power supply and multi-node energy transfer capability, we have investigated the charging scheduling problem, using multiple MCs in on-demand WRSNs. First, we have presented a novel network partitioning approach to distribute the MCs in a load balanced manner. Next, with an objective of avoiding the unfair charging response by the MCs, we have established a mathematical formulation to calculate the adaptive recharging thresholds for the nodes. Finally, we have adopted the Mamdani Fuzzy Inference System which determines the charging schedules of the nodes by wisely integrating multiple attributes, such as residual energy, distance to MC, critical node density, and energy consumption rate.

Rigorous simulations have been carried out to prove the supremacy of the proposed FLCSD scheme over the state-of the- art schemes. The results have revealed that as compared to the state-of-the-art schemes, the FLCSD provides better stability by increasing the number of survived nodes and boosts the charging performance by achieving both low average charging latency and high energy usage efficiency. The simulation results are also validated through the statistical ANOVA test. However, the proposed scheme has not considered the coordination among the MCs for recharging the nodes as well as the MCs, which may further improve the charging performance, especially in large-scale WRSNs. We will try to incorporate this feature in our future work by exploring new multi-attribute decision making methods.

5. REFERENCE

[1] M. Ayaz, M. Ammad-uddin, I. Baig et al., "Wireless sensors civil applications, prototypes, and future integration possibilities: A review," IEEE Sensors Journal, vol. 18, no. 1, pp. 4–30, 2017.

[2] P. Kar and S. Misra, "Reliable and efficient data acquisition in wireless sensor networks in the presence of transfaulty nodes," IEEE Transactions on Network and Service Management, vol. 13, no. 1, pp. 99–112, 2016.

[3] S. Gao, H. Zhang, and S. K. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," IEEE Transactions on Mobile Computing, vol. 10, no. 4, pp. 592–608, 2010.

[4] B. Fateh and M. Govindarasu, "Joint scheduling of tasks and messages for energy minimization in interference-aware real-time sensor networks," IEEE transactions on mobile computing, vol. 14, no. 1, pp. 86–98, 2013.

[5] S. Guo, Y. Shi, Y. Yang, and B. Xiao, "Energy efficiency maximization in mobile wireless energy harvesting sensor networks," IEEE Transactions on Mobile Computing, vol. 17, no. 7, pp. 1524–1537, 2017.

[6] A. Mehrabi and K. Kim, "General framework for network throughput maximization in sink-based energy harvesting wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 16, no. 7, pp. 1881–1896, 2017.

[7] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Solja˘ci´c, "Wireless power transfer via strongly coupled magnetic resonances," science, vol. 317, no. 5834, pp. 83–86, 2007.

[8] K. Kang, Y. S. Meng, J. Br´eger, C. P. Grey, and G. Ceder, "Electrodes with high power and high capacity for rechargeable lithium batteries," Science, vol. 311, no. 5763, pp. 977–980, 2006.

[9] M. Hu, Z. Chen, K. Peng, X. Ma, P. Zhou, and J. Liu, "Periodic charging for wireless sensor networks with multiple portable chargers," IEEE Access, vol. 7, pp. 2612–2623, 2018.

[10] L. Xie, Y. Shi, Y. T. Hou, W. Lou, H. D. Sherali, and S. F. Midkiff, "Multi-node wireless energy charging in sensor networks," IEEE/ACM Transactions on Networking (ToN), vol. 23, no. 2, pp. 437– 450, 2015.