ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

Mobility Management In Wireless Sensor Network

Megha M. Chaitanya[1], Amit R. Sarkar[2]

ME Student, Computer Science and Engineering, College of Engineering Pandharpur (SVERI), Pandharpur, India[1]

Professor, Computer Science and Engineering, College of Engineering Pandharpur (SVERI), Pandharpur, India[2]

Abstract: Wireless sensor network consist of sensors which senses some physical phenomenon then information is gathered and processed to get result. There are various issues regarding the mobility management such as coverage, connectivity, energy consumption, Location management. Wireless sensor networks are being used in different area such as in military, wildlife tracking, Industry security, disaster management. In many such tasks, node localization is very important. Node localization is required to report the origin of events, assist group querying of sensors, routing and to answer questions on the network coverage. So, one of the fundamental challenges in wireless sensor network is node localization. In real world if we want to implement some experiments with Wireless Sensor Network then are various practical difficulties. So to analysing its performance simulation essential for the study of WSNs. The ns-2 network simulator is one of the most widely used tool to investigate the characteristics of WSNs. This paper presents an extension of ns-2.

Keywords: WSN, Simulator, Localization, Ns-2, Otcl.

I. INTRODUCTION

There are various issues in mobility management in wireless sensor network such as location management, coverage, connectivity, energy consumption, etc. Among these issues location management is very important issue so I have focused on location management. Location information plays a critical role in wireless sensor networks (WSN). Most of the WSN applications and techniques require that the positions of the sensor nodes be determined. Localization algorithms (e.g. [1-5]) follow several approaches to estimate positions of sensor nodes.
There are two types of node one special node called beacons, which know their own location, through a GPS receiver or manual configuration. The other nodes that do not know their location, sometimes referred to as unknowns. Unknown nodes use different techniques to compute their own position based on the location information of the beacons and the measured distance to these beacons. Once the unknown node obtains its position, it could act as a reference for other unknowns.

In WSN Testbeds[6],[7] are used to study various aspects of WSNs. However, because of some practical difficulties in setting up a WSN and analysing its performance, the use of simulation is essential for the study of WSNs. Simulation is widely used in system modelling for applications ranging from engineering research, business analysis, manufacturing planning and biological science experimentation [8]. NS-2 is an open-source event-driven simulator designed specifically for research in networks. Ns-2 was developed in C++ and uses Object-oriented Tool command Language (OTcl) as a configuration and script interface (i.e., a front-end). Each language has two types of classes. The first type includes the standalone C++ and OTcl classes that are not linked together. The second type includes classes which are linked between the two languages. These C++ and OTcl classes are respectively called, "compiled hierarchy" and "interpreted hierarchy". These two hierarchies are linked together using an OTcl/C++ interface called TclCL [12].

In this paper, ns-2 is extended to simulate localization systems in WSNs. New modules are added and a base class, called "Position", is created. Position enables the sensor nodes to estimate their position using the general multilateration method.. A simulation is conducted to evaluate the performance of Position.

II. LOCALIZATION SYSTEM

Localization systems consist of three major components: distance estimation, position computation and a localization algorithm [14], as shown in Figure 1.

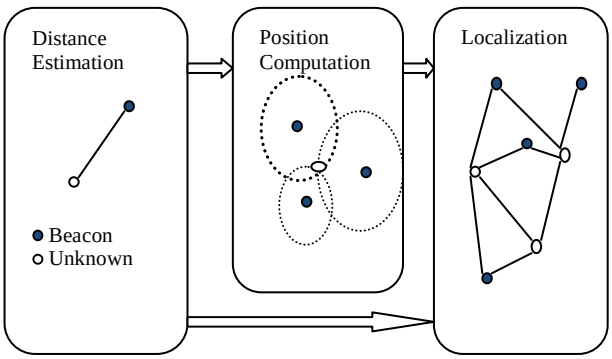

Fig. 1. Localization system.

A. Distance Estimation

This component is responsible for determining the physical relationship between two nodes. That information can later be used to compute a node's location. Different techniques can be used for this purpose, such as RSS, ToA, time difference of arrival, angle of arrival or round-trip time. These techniques have been investigated in more

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

detail in [15]. In proposed algorithm RSS (Received signal strength) is used.

B. Position Computation

This component is responsible for computing the position of a node based on available information about the distance estimated from the previous component and position of references. Techniques used in this component include triangulation [3], trilateration [16] and multilateration [17]. Among all above mentioned techniques the multilateration technique is very useful technique so this technique is selected.

C. Localization Algorithm

This is the main component of a localization system. It manipulates the available information to enable most or all of the nodes of the WSN to estimate their position.

III. LOCALIZATION EXTENSION

In NS-2 there are several flexible modules for energy-constrained wireless ad-hoc networks, which encourage researchers to use Ns-2 to investigate the characteristics of WSNs. However, to implement and evaluate localization algorithms, ns-2 should be extended and new modules should be added.

There are some standalone classes, which are MMSE, Position and YourPosition classes. These classes are created in the C++ domain. There are compiled hierarchy classes, which are LocDisApp, LocReqAgent and LocResAgent classes. In order to be able to access the newly compiled hierarchy classes LocDisApp, LocReqAgent and LocResAgent from the OTcl domain, these classes were mapped and linked to corresponding OTcl classes, which are Application/LocDis, Agent/LocReq and Agent/LocRes, respectively.

A. *Class Hierarchy*

The Doxygen documentation system [19] was used to illustrate the class hierarchy of the new classes.

1) MMSE Class:

This class is responsible for all the mathematical matrices operations required to solve .In this class optimized methods dedicated mainly to MMSE were implemented. These optimized methods require less computation and shorter execution time.

2) Position Class:

Position class represents a general localization method using the basic multilateration method for position computation and RSS for distance estimation. This method uses all of the available references, does not distinguish between beacons and references, does not weigh the references used and performs the estimation only once. This class is the base class for any localization algorithms implemented. The Position class uses an array of this structure to store the location information (location and distance) of neighbouring references. The Location class represents the X, Y and Z coordination of sensor nodes.

4) LocReqAgent and LocResAgent Classes:

These classes are derived from the Agent class. LocReqAgent constructs and broadcasts a "location request" packet. This agent should be attached only to unknown nodes because beacons already know their location. LocResAgent is responsible for handling the packets received. This agent should be attached to all nodes (unknowns and beacons). Two types of packets could be received. The first is a "location request" packet. If this type of packet is received by a beacon or reference node it constructs a "location response" packet that includes location information and then sends it to the requesting node. Unknown nodes receiving this type of packet simply deallocate it. The second is a "location response" packet. The requesting node that receives this packet sends it to the application layer (LocDisApp), which processes the included location information to estimate the node's position.

5) LocDisApp Class:

The LocDisApp class is derived from the Application class. Each node in the network uses an object from this class by attaching it to its agent(s). The LocDisApp class performs several functions, such as invoking the broadcast method of LocReqAgent periodically to broadcast a "location request" packet, processing the received "location response" packet and estimating the node location. This class collaborates with several classes, which are the three timers, the two agents, Location and Position classes. It uses the hdr_locres header structure to gain access to the received "location response" packets in order to process the included location information and estimate the node location. LocDisApp uses a vector of class ResData, which is used to store the location information received from neighbouring references.

6) The Timer Classes:

There are three types of timer classes, ReqTimer, EstimateTimer and OutputTimer classes. These Classes are derived from the TimerHandler class. These timer classes collaborate with LocDisApp to schedule several tasks during the run time. The ReqTimer is used to moderate how frequently sensor nodes broadcast a "location request" packet.

After sending the "location request" packet, the EstimateTimer is used to schedule the estimation process after a specific delay, which is required to give "location response" packets enough time to receive information from neighbouring references. The OutputTimer is used to schedule the action of recording the result (such as location error, number of references used and remaining energy) to the trace file.

7) ResData Class:

This class is responsible for storing and retrieving the location information included in the "location response" packets received from the neighbouring references. This information includes the address, the location and the power with which the packet is received.

Implementing a new localization algorithm could require adding extra attributes to this class. Introduced the term "probability of accuracy" of the reference nodes, which requires selecting a subset of references that will be used

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

to estimate node's position.

B. *Node Configuration*

At the beginning of the simulation there are only two types of nodes, beacons and unknowns. Each of them has a different configuration, as shown in Figure. Beacons already know their location, so they should be attached only with LocResAgent. Unknowns should be attached with LocReqAgent in order to allow them to broadcast location request messages, and they should also be attached with LocResAgent to handle the recipient packets (which are "location request" and "location response" packets). These two types of agents should be attached to LocDisApp.

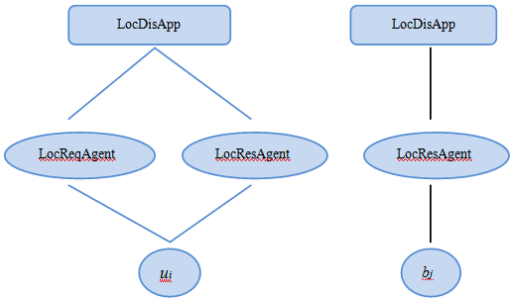

Fig 4. Node configuration

Before creating the nodes their configuration (e.g. the routing protocol, MAC type, etc.) should be specified. In addition, the attribute (either BEACON or UNKNOWN) of the nodes should also be specified; this can be done with the help of the command "node-config".

IV. PROPOSED LOCALIZATION PROCEDURE

Most of the Existing algorithm relays only on high number of references to estimate the position of node. These approaches improve the accuracy but by using more number of references requires more computation more space of memory and consumes more energy. So these approaches are infeasible for resource constrained WSNs. Using only a subset of references with a chance of higher accuracy could help to overcome the problem associated with using all the available references Moreover following this approach (selecting only subset of references) will help to obtain several design goals such as accuracy, Robustness, simplicity, security, energy efficiency, time consumption .The complete procedure of the localization process is as follows:

BEGIN

1) Initialisation

-If(final=true) then end

-Unknown node broadcasts "location request "message.

-Neighbouring beacon or references (Ri) broadcasts "Location response message.

2) Initial Position Estimation

- Select a subset of references (Si) from R(i).
- Measure distance to references.
- Apply MMSE to determine initial position.

3) Refined Position Estimation

- Check for any enhancement
- Estimate the refined position

4) Position updating

- Node updates the new position
- Final=true.

End

V. OUTPUT MANIPULATION

After simulation, ns-2 outputs either text-based or animation-based simulation results. Several tools could be used to interpret these results graphically or interactively. The Network Animator (NAM) [9] is an animation tool that uses the animation-based results to view the network simulation traces and real-world packet traces. NAM supports topology layout, packet level animation and various data inspection tools. Text-based results consist of a lot of details on events that occur in the network, such as the sending or receiving of packets, and the movement and remaining energy levels of nodes. A new method was written that records the localization-related information in the trace file. Several tools can be used to extract this location information into a different format such as graphical or animation.

VI. RESULT ANALYSIS

Figure 6 shows the animation file in which there are two types of node the green nodes are the unknown nodes and the brown nodes are nodes either beacon or references. It shows x and y coordinated of the node.

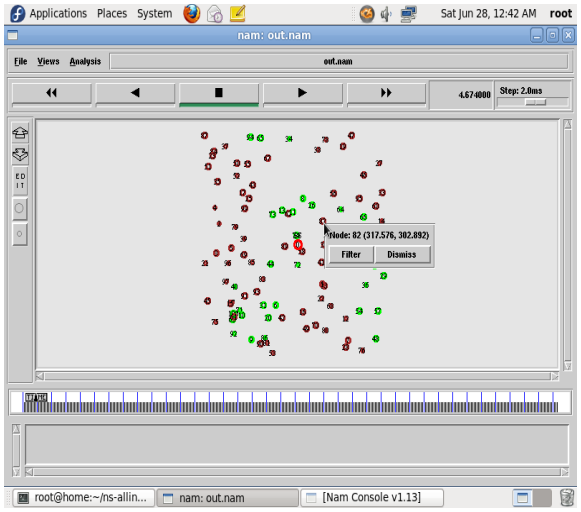

Fig 6.Location of node is determined

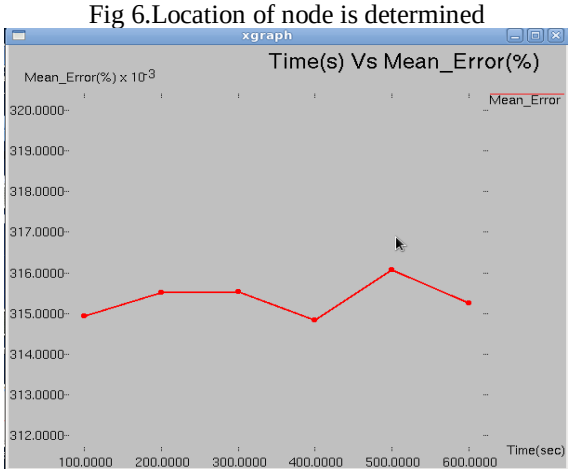

Fig 7. Time vs Mean_Error

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

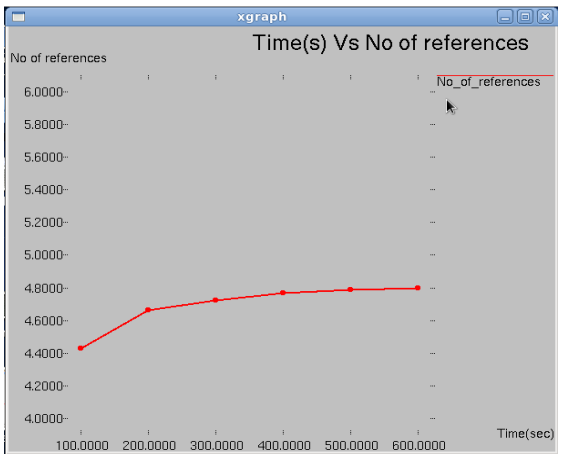

Fig 8. Time Vs. No. of References

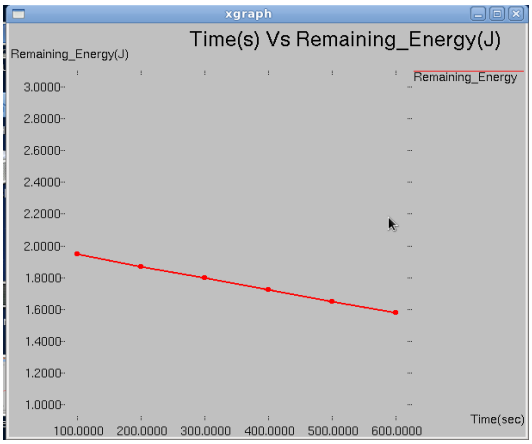

Fig 9.Time Vs.Remaining_Energy

VII. CONCLUSION

It can be concluded that with the help of NS2 the location of the node is determined which is not equipped with the GPS receiver. This paper presents an extension to ns-2. This paper shows that the localization technique used is cost-effective, requires less number of references, less time and the mean error is minimized. This technique is implemented where multiple sensor nodes are exists.

VIII. FUTURE WORK

Future research is required to achieve more energy efficiency. Implementing a secure localization system is another challenging area. Performance must be evaluated in both terms of overheads and resistance to attack. This thesis accessed the performance evaluation of the proposed algorithm by network simulation. An extension to this work is actual implementation of proposed algorithm and field testing would be a challenging task.

REFERENCES

[1] A. M. Abu-Mahfouz and G. P. Hancke, "An efficient distributed localization algorithm for wireless sensor networks: Based on smart reference-selection method," Submitted for Publication, 2011.

[2] S. Chinnappen-Rimer and G. P. Hancke, "Perimeter echo algorithm for network localization," in Proceedings of the IEEE AFRICON 2009, September 23 - 25, Nairobi, Kenya, 2009, pp. 1-5.

[3] K. Y. Cheng, V. Tam and K. S. Lui, "Improving aps with anchor selection in anisotropic sensor networks," in Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services — ICAS-ICNS '05, October 23-28, Papeete, Tahiti, 2005, pp. 49-54.

[4] D. Lieckfeldt, J. You and D. Timmermann, "An algorithm for distributed beacon selection," in Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communication — PerCom '08, March 17-21, Hong Kong, China, 2008, pp. 318-323.

[5] S. Han, S. Lee, S. Lee, J. Park and S. Park, "Node distribution-based localization for large-scale wireless sensor networks," Wireless Networks, vol. 16, no. 5, pp. 1389-1406. 2010.

[6] A. M. Abu-Mahfouz, L. P. Steyn, S. J. Isaac and G. P. Hancke, "Multi-level infrastructure of interconnected testbeds of large scale wireless sensor network (MI2T-WSN)," in Proceedings of the 11th International Conferences on Wireless Networks, July 16-19, Las Vegas, USA, 2012, .

[7] L. P. Steyn and G. P. Hancke, "A survey of wireless sensor network testbeds," in Proceedings of the IEEE AFRICON, September 13-15, Livingstone, Zambia, 2011, pp. 1-6.

[8] T. Issariyakul and E. Hossain, Introduction to Network Simulator (NS2). New York, NY, USA: Springer, 2009.

[9] "The network simulator - ns-2," 12 February 2010, online available: ttp://nsnam.isi.edu/nsnam/index.php/User_Information.

[10] "OMNET++ simulator," 09 March 2011, online available: http://www.omnetpp.org/.

[11] P. Levis, N. Lee, M. Welsh and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in Proceedings of the 1st International Conference on Embedded Networked Sensor System — SenSys '03, November 05-07, Los Angeles, CA, USA, 2003, pp. 126-137.

[12] K. Fall and K. Varadhan, "The Ns Manual," 9 May 2010.

[13] P. Morávek, "Ns-2 simulator capabilities in nodes localization in wireless networks," 2009

[14] A. Boukerche, H. Oliveira, E. F. Nakamura and A. A. Loureiro, "Secure localization algorithms for wireless sensor networks," IEEE Communications Magazine, vol. 46, pp. 96-101, 2008.

[15] A. M. Abu-Mahfouz and G. P. Hancke, "Distance bounding: A practical security solution for real-time location systems?" IEEE Transactions on Industrial Informatics, Accepted for publication.

[16] S. Tian, X. Zhang, X. Wang, P. Sun and H. Zhang, "A selective anchor node localization algorithm for wireless sensor networks," in Proceedings of the International Conference on Convergence Information Technology — ICCIT '07, November 21-23, Gyeongju, Korea, 2007, pp. 358-362.

[17] J. Liu, Y. Zhang and F. Zhao, "Robust distributed node localization with error management," in Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, May 22-25, Florence, Italy, 2006, pp. 250-261.

[18] A. Savvides, C. C. Han and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking — MobiCom '01, July 16-21, Rome, Italy, 2001, pp. 166-179.

[19] "The doxygen documentation system," 12 October 2010.

BIOGRAPHY

Megha M. Chaitanya has obtained her B.E. degree in Computer Science and Engineering from college of Engineering, Pandharpur.Solapur, Maharashtra, India. She is currently pursuing M.E. in CSE at College Of Engineering Pandharpur, Solapur, Maharashtra, India. Her area interest accumulated in Networking, Distributed system and Image Processing.