Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 708-714

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Implementation of Stationary Models for Localization in Wireless Sensor Network using ns3

[1]M. Arun, [2]M. Brindha and [3]Dr. P.T. Vanathi

[1]*Research Scholar, Department of Electronics and Communication Engineering, Karpagam University, Coimbatore, TamilNadu, India*
[2]*Alumni, Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, TamilNadu, India.*
[3]*Associate Professor, Department of Electronics and Communication Engineering PSG College of Technology, Coimbatore, TamilNadu, India*

ARTICLE INFO

Article history:
Received 3 June 2015
Received in revised form 17 June 2015
Accepted 1 August 2015
Available online 15 October 2015

Keywords:

ABSTRACT

Wireless sensor networks (WSNs) are usually randomly deployed in region of interest. As a result, algorithms that can compute the location of sensor nodes within a WSN are needed. In recent years, several localization algorithms have been proposed for stationary WSNs. However, most studies only provide simulation results using familiar tools such as mat lab, ns2, qualnet etc and most algorithms have never been implemented using ns3. In this paper, we implement stationary models for localization in stationary WSNs using ns3. The results obtained show that the implementation results are consistent with the simulation results.

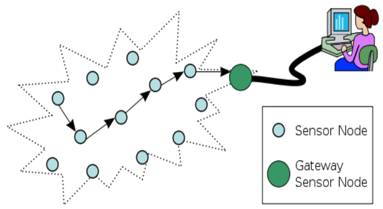
© 2015 AENSI Publisher All rights reserved.

To Cite This Article: M. Arun, M. Brindha and Dr. P.T. Vanathi., Implementation of Stationary Models for Localization in Wireless Sensor Network using ns3. ***Aust. J. Basic & Appl. Sci.,*** *9(27): 708-714, 2015*

INTRODUCTION

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location (http://en.wikipedia.org/wiki/).

Sensors connect to controllers and processing stations directly (e.g., using local area networks), an increasing number of sensors communicate the collected data wirelessly to a centralized processing station. A wireless sensor has not only a sensing component, but also on-board processing, communication, and storage capabilities. With these enhancements, a sensor node is often not only responsible for data collection, but also for in-network analysis, correlation, and fusion of its own sensor data and data from other sensor nodes. When many sensors cooperatively monitor large physical environments, they form a wireless sensor network. Sensor nodes communicate not only with each other but also with a base station (BS) using their wireless radios, allowing them to disseminate their sensor data to remote processing, visualization, analysis, and storage systems.



WSNs are commonly designed with an objective of gathering information from the surrounding area of deployment to study and/or monitor a variety of phenomena. The information gathered by each sensor node is required to be sent to a workstation through a central sensor node, known as the *base station*, to be accessed by the user and for further processing. The initial reason behind the development of WSNs was the military need for applications such as battlefield surveillance. Besides military usage, WSNs are also used in a variety of Localizing and tracking moving stimuli or objects is an essential capability for a sensor network in many practical applications. In our project we focus on localization and its problems.

Node localization, which is determining the physical or relative location of a sensor node, is extremely important in almost any WSN application (Holger karl, Andreas Willig, 2005). It can be used for identifying the source transmitting data, supporting certain routing protocols, location-aware services, etc.

Corresponding Author: M. Arun, Research Scholar, Department of Electronics and Communication Engineering, Karpagam University, Coimbatore, TamilNadu, India
E-mail:arunkset@gmail.com

Sensor nodes with known location information are called ‘Anchor nodes’ or simply anchor for short. Typically, anchor nodes obtain their location information by using a global positioning system (GPS), or by manually being placed at defined coordinates. Likewise, sensor nodes with unknown location information are called ‘Non-Anchor nodes’ or simply (sensor) nodes.

The rest of this paper is organized as follows. Section 2 briefs about Network simulator 3 (ns3) features and its advantages over ns2. Then, Section 3 presents the models used for network deployment. Section 4 focuses on models for localization which is the goal our paper. Section 5 presents the implementation results comparison of grid position allocator with various stationary models. Finally, the paper is concluded in Section 6.

I. Network Simulator 3:

Network Simulator 3 (ns3) is a discrete-event network simulator, targeted primarily for research and educational use. It is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

The goal of the ns-3 project is to develop a preferred, open simulation environment for networking research: it should be aligned with the simulation needs of modern networking research and should encourage community contribution, peer review, and validation of the software.

The ns-3 software infrastructure encourages the development of simulation models which are sufficiently realistic to allow ns-3 to be used as a real-time network emulator, interconnected with the real world and which allows many existing real-world protocol implementations to be reused within ns-3.

The ns-3 simulation core supports research on both IP and non-IP based networks. However, the large majority of its users focuses on wireless/IP simulations which involve models for Wi-Fi, WiMAX, or LTE for layers 1 and 2 and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications.

Ns-3 also supports a real-time scheduler that facilitates a number of "simulation-in-the-loop" use cases for interacting with real systems. For instance, users can emit and receive ns-3-generated packets on real network devices, and ns-3 can serve as an interconnection framework to add link effects between virtual machines.

Ns-3 is built using C++ and Python with scripting capability. ns-3 is not backward compatible with ns-2, instead, it is built from the scratch. It is written in C++ and Python Programming Language can be optionally used as an interface. It has extensive doxygen documentation, which is available on the Internet.

Ns-2 is often criticized because modelling is a very complex and time-consuming task, since it has no GUI and one needs to learn scripting language, queuing theory and modelling techniques. Also, of late, there have been complaints that results are not consistent (probably because of continuous changes in the code base) and that certain protocols are replete with bugs.

Ns-3 is trying to tackle problems present in ns-2 such as lack of memory management, coupling between different models (C++ weak base class problem) etc.viz.,

i. The total computation time required to run a simulation scales better in ns-3 than ns-2.

ii. ns-3 has good Memory usage and quite fast in computation time when compared to ns-2

iii. In ns-2, bi-language system make debugging complex but for ns-3 only knowledge of C++ is enough (single-language architecture is more robust in the long term).

iv. ns-3 has an emulation mode, which allows for the integration with real networks and well suited for sensor networks than ns-2

Due to the advancement of ns-3 over ns-2, we choose ns-3 for developing localization in WSN.

II. Network Deployment Models:

The ns-3 simulator is a discrete-event network simulator targeted primarily for research and educational use and it is not an extension of ns-2. The ns-3 library is split across many modules organized under the Modules tab. Out of these modules ,the one used in this paper is the Mobility module. The mobility support includes:

i. a set of mobility models which are used to track and maintain the "current" Cartesian position and speed of an object.

ii. a "course change notifier" trace source which can be used to register listeners to the course changes of a mobility model

iii. a number of helper classes which are used to place nodes and setup mobility models

This model comprises of more than 20 classes. Like, in C++ each class has member functions, attributes, parameters etc. In these classes we can classify classes as that are used for allocating position of nodes i.e. deployment of nodes and that are used for mobility of nodes in wireless sensor nodes.

Some of the position allocators are as follows:

a) Box,
b) Building,
c) List Position Allocator,
d) Grid Position Allocator,
e) UniformDisc Position Allocator,
f) RandomDisc Position Allocator,
g) RandomBox Position Allocator,
h) RandomRoom Position Allocator,
i) RandomBuilding Position Allocator,
j) RandomRectangle Position Allocator.

And some of the mobility models are as follows:

Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 708-714

k) Constant Position Mobility Model,
l) Constant Velocity Mobility Model,
m) Constant Acceleration Mobility Model,
n) Random Walk2d Mobility Model,
o) Random Waypoint Mobility Model,
p) Hierarchical Mobility Model,
q) Random Direction 2d mobility model,
r) Way Point Mobility Model.

For stationary nodes, Constant (Position /Velocity/Acceleration) Mobility model and for deployment of nodes Position allocator is used. It has no attribute or trace source. Using this model we can change the remaining parameters such as speed, velocity, position.

In this paper, we use some of the above models for network deployment.

A. Grid Allocation of Stationary Nodes:

Allocate positions on a rectangular 2d grid. The attributes of grid position allocator are :

- GridWidth: The number of objects laid out on a line.
- MinX/ MinY: The x/ y coordinate where the grid starts.
- DeltaX/ DeltaY: The x/y space between objects.
- LayoutType: The type of layout.

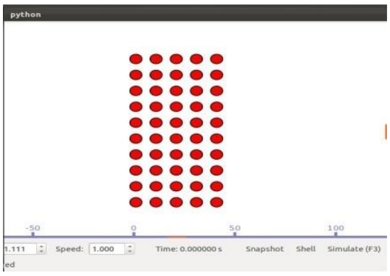


Fig. 1: Grid Position allocation – width 5.

The grid can be formed by varying its attributes as per the requirements.
In Fig.1, the grid width is set as 5 and the delta (x, y) value is set as (10,10) and in fig 4.2 the width is set to 10 without changing the other attributes and number nodes is 50.

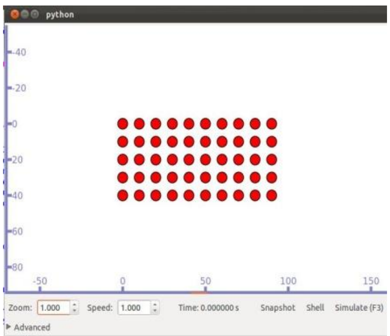


Fig. 2: Grid Position allocation – width 10.

B. Random Allocation of stationary nodes:

a) Random Disc Position Allocator:

Allocate random positions within a disc according to a given distribution for the polar coordinates of each node with respect to the provided centre of the disc. The attributes are

- Theta: A random variable which represents the angle (gradients) of a position in a random disc.
- Rho: A random variable which represents the radius of a position in a random disc.
- X: The x co-ordinate of the centre of the random position disc.
- Y: The y co-ordinate of the centre of the random position disc.

In fig 3, the random allocation of 9 nodes with (x,y) min & max value set as (0,0) & (100,100) respectively and Rho as 30and the theta value is taken as default one.

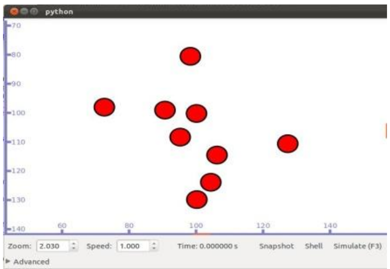


Fig. 3: Random Disc Position allocation.

b) Random Box Position Allocator:

Allocate random positions within a 3D box according to a set of three random variables. The attributes are

- X/Y/Z: A random variable which represents the x/y/z coordinates of a position in a random box.

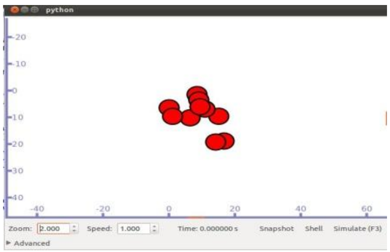


Fig. 4: Random Box Position allocation.

In fig 4, the random box allocation with (x,y,z) is set as min (0,0,0) & max (20,20,20).

c) Random Rectangle Position Allocator

Allocate random positions within a rectangle according to a pair of random variables. The attributes are:

- X: A random variable which represents the x coordinates of a position in a random rectangle.
- Y: A random variable which represents the y coordinates of a position in a random rectangle.

The Random Rectangle Position allocation model of 10 nodes with (x,y) set as min (0,0) & max(20,20) is shown in fig 5.

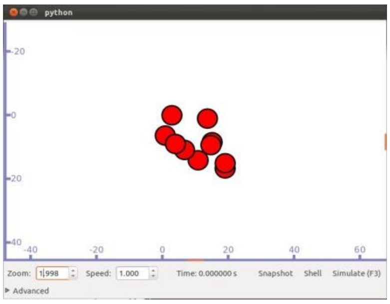


Fig. 5: Random Rectangle Position allocation.

C) Grid Allocation of Mobile Nodes:

a) Random Walk2D Mobility model:

Each instance moves with a speed and direction chosen at random with the user-provided random variables until either a fixed distance has been walked or until a fixed amount of time. If we hit one of the boundaries (specified by a rectangle), of the model, we rebound on the boundary with a reflexive angle and speed. This model is often identified as a Brownian motion model. The attributes are :

- Bounds: Bounds of the area to cruise.
- Time: Change current direction and speed after moving for this delay.
- Distance: Change current direction and speed after moving for this distance.
- Mode: It indicates the condition used to change the current speed and direction
- Direction: A random variable used to pick the direction (gradients).
- Speed: A random variable used to pick the speed (m/s).

These attributes have no trace sources. The attributes that have trace sources are

- Position: The current position of the mobility model.
- Velocity: The current velocity of the mobility model, and the trace sources defines here is Course Change: The value of the position and/or velocity vector changed.

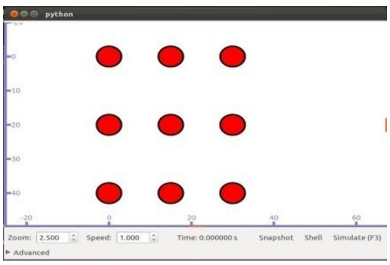


Fig. 6.a: Grid with Random walk 2d model.

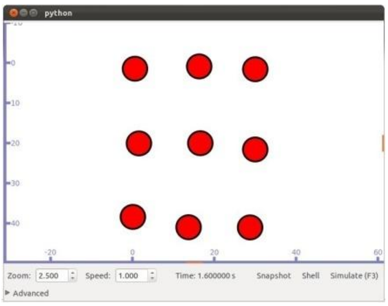


Fig. 6.b: Grid with Random walk 2d model.

The fig 6.a shows the grid deployment and fig 6.b shows the mobility of nodes in that grid topology using Randomwalk2d mobility model

D) Random Allocation of Mobile Nodes:

a) Random Disc allocator with RandomWalk2d mobility:

The Random Disc Position Allocator is set with initial value (x,y) as (0,0) and Rho as (min, max) = (0,30) and the Random walk2d mobility model with bounds as Rectangle value (-50,50,-50,50) in fig 7

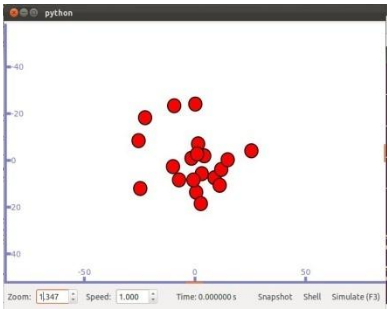


Fig. 7a: Random Disc with Random walk.

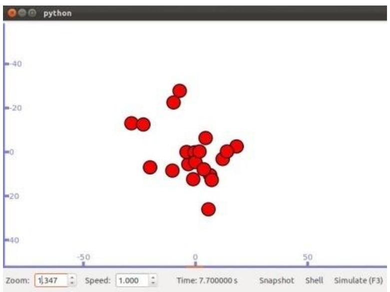


Fig. 7b: Random Disc with Random walk 2d model.

b) Random Rectangle Allocator with RandomWalk2D Mobility:

The Random Rectangle Position Allocator is set with value of (x,y) as min(0,0) & max(20,20) and the Random Walk2D mobility model with speed of 1 and bound of (0,200,0,200) in fig 8.

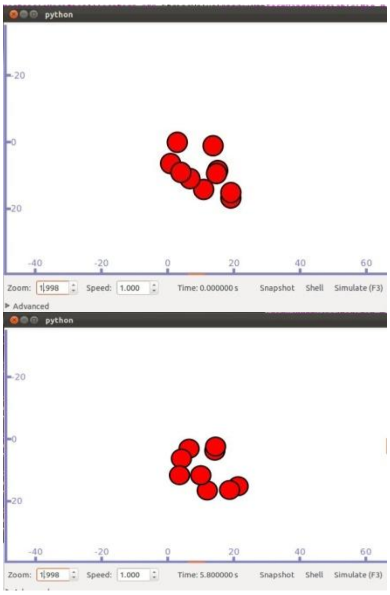


Fig. 8: Random Rectangle with Random walk.

Similarly, all combination of Random Allocation and Random Mobility model will takes place.

III. Localization:

Node localization, which is determining the physical or relative location of a sensor node, is extremely important in almost any WSN application. The position of nodes in various topologies is found. The source node sends packets to the sink nodes as it founds the position of the nodes.

A. Grid Allocation of Stationary Nodes

The constant mobility model is applied to the the grid position allocator model.then the location of nodes are found as in fig 9, and the packets are transmitted to the to all other nodes from node 1 as per the data given as shown in the fig 10.

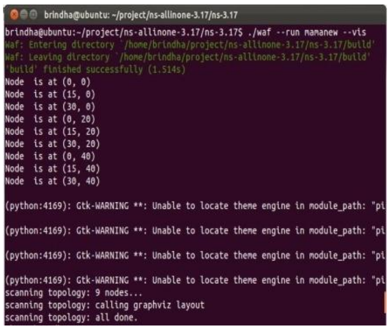


Fig. 9: Position of stationary nodes with Grid.

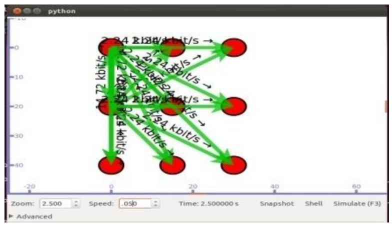


Fig 10: Transmission between stationary nodes.

B. Random Allocation of Mobile Nodes:

The random mobility model is applied to the grid position allocator model. Then the positions of nodes are found and the packets are transmitted from one node to the remaining nodes as shown in fig 11.

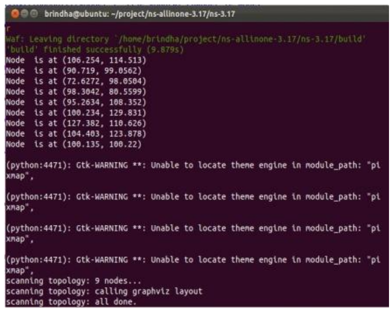


Fig. 11: Position of Mobile Nodes with Grid Allocation.

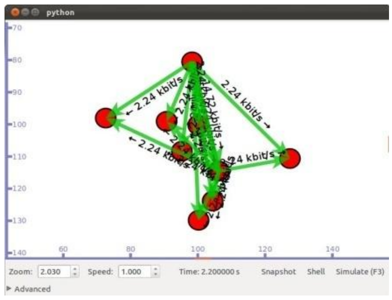


Fig. 12: Transmission between Mobile Nodes.

III. Results:

In Grid Allocation the attribute varied for analysis are grid width, number of nodes , Layout type etc...and the position of each node is determined using the models and the accuracy in position for each attribute change is analysed.

The position of nodes changes with respect to the attribute change of Grid Model. In this case , the degree of accuracy can't be determined since the position of nodes in Grid is uniform randomly allocated ,similar to matrix pixel allocation in an image. In order to analyse clearly we go for analysing Random Position Allocator .We had chosen Random Disc Allocation for analysing by varying the attributes such as number of nodes, Rho, dimensions etc...

Table 1: Position of Nodes with Varying Grid Width.

Grid: No. Of Nodes -10, Layout type – Row First, Delta x=10 , Delta y=10			
Node number	Node Position		
	Width 3	Width 5	Width 10
0	(0, 0)	(0, 0)	(0, 0)
2	(30, 0)	(30, 0)	(30, 0)
5	(30, 20)	(0, 20)	(75, 0)
9	(0, 60)	(60, 0)	(135, 0)

Table 2: Position of Nodes with Varying Grid Layout Type.

Grid: No. Of Nodes -10, Width – 3 Delta x=10 , Delta y=10		
Node number	Layout type (Node Position)	
	Row First	Column First
0	(0, 0)	(0, 0)
2	(30, 0)	(0, 40)
5	(75, 0)	(0, 100)
9	(135, 0)	(0, 180)

Table 3: Position of Nodes with Varying Random Disc dimension.

Random Disc: No. Of Nodes -10, Rho - 30		
Node number	x=100 y=100	x=0 y=0
	Node Position	
0	(104.371, 100.233)	(6.25388, 14.5126)
2	(99.3966, 101.205)	(-27.3728, -1.94964)
5	(93.9176, 108.08)	(0.234265, 29.8306)
9	(77.9463, 97.9457)	(0.134888, 0.219648)

Table 4: Position of Nodes with varying Random Disc Rho.

Random Disc: No. Of Nodes -10, x=100,y=100		
Node Number	Rho 50	Rho 70
0 2 5 9	(107.285, 100.389) (98.9944, 102.009) (89.8627, 113.466) (63.2439, 96.5762)	(110.199, 100.544) (98.5921, 102.812) (85.8078, 118.853) (48.5414, 95.2066)

Table 5: Position of nodes with varying number of nodes.

Random Disc: Rho 30 x=100,y=100		
Node Number	No. Of Nodes -10	No. Of Nodes -100
0	(104.371, 100.233)	(88.398, 91.909)
3	(87.5007, 125.767)	(100.48,118.179)
7	(84.1212, 113.077)	(126.35, 98.496)

From the above table analysis it is clear that at random disc position allocator, the position of each node varies randomly as we change any of the attributes in it. The position of nodes in WSN has been found i.e., localization of nodes is done using various models in ns3.

IV. Conclusion:

The goal of this paper is to implement localization of stationary nodes in Wireless Sensor Networks using the Network Simulator3 (ns3), i.e., finding the position of each node present in the network. We have deployed nodes using various Position Allocator Model viz., Grid Position Allocator, Random Disc Position Allocator, Random Rectangle Position Allocator, Random Box Position Allocator , etc. along with the Mobility model viz., Constant Position Mobility , Constant Velocity Mobility, Constant Acceleration Mobility models for stationary nodes and Random Walk2dMobility , Random Waypoint Mobility etc. for mobile Nodes.

Our main goal is to find location of nodes in stationary node and the results are verified by varying number of nodes , range and various parameter with respect to the model used is done and verified.

The future work is to stimulate various models in ns3 and the most accurate location finding model independent of any parameter is found and compared with localization algorithms for Wireless Sensor Networks implemented in other tools also.

REFERENCES

http://en.wikipedia.org/wiki/Wireless sensornetwork

Waltenegus Dargie, Christian Poellabauer, 2010. "*Fundamentals of Wireless sensor networks:*

Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 708-714

Theory and Practice", A John Wiley and Sons, Ltd., Publication.

http://www.nsnam.org/doxygen-release/classns3_1_1_box.html

Feng zhao, Leonidas J. Guibas, *2004. "Wireless sensor networks: an information processing approach",* Morgan Kaufmann Publication.

Holger karl, Andreas Willig, 2005. "*Protocols and architectures for wireless sensor networks",* John Wiley & Sons Ltd, Publication.

Ahmed Al-Odhari, Shafagh Alikhani, Ahmed Dirie, Hamed Omid Beki, Laryssa Sirko, Marc St-Hilaire, and Thomas Kunz, 2011. "Implementation of the CCA-MAP Localization Algorithm on a Wireless Sensor Network Testbed*" IEEE 978-1-4577-9538-2/11/$26.00.*

Chih-Yung Chang, *Associate Member, IEEE*, Chih-Yu Lin and Chao-Tsun Chang, 2012. "Tone-Based Localization for Distinguishing Relative Locations in Wireless Sensor Networks" *IEEE SENSORS JOURNAL, 12-5*.

Javad Rezazadeh Marjan Moradi Abdul Samad Ismail, 2011. "Efficient localization via Middle-node cooperation in wireless sensor networks " *International Conference on Electrical, Control and Computer Engineering* Pahang, Malaysia, 21-22.

Chen Sibo, Lin Zhiwei, Huang Hong, 2011. "Wireless Sensor Network Node Localization Algorithm based on Adjacent Node Relationship " *IEEE 978-1-61284-486-2/111*.

Demartines, P. and J. Hrault, 1997. "Curvilinear component analysis: A self organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, 8(1): 148–154.

Pierre Demartines and Jeanny H´erault, 1995. "CCA : Curvilinear Component Analysis" *Proc . of 15th workshop GRETSI* , Juan-Les-Plns, France,.

Pierre Demartines and J. H´erault, 1997. "Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets*" IEEE Transactions on Neural Networks, 8-1.*

Li, L. and T. Kunz, 2007. "Localization applying an efficient neural network mapping," in *First International Conference on Autonomic Computing and Communication Systems*, 1.