Mobility Support Enhancements for a Wireless Sensor Network Framework

Raphael A. Camponogara Viera,Luana Palma, Leonardo Londero de Oliveiraand João Baptista Martins
{raphaelviera,luanapalma,leonardo}@mail.ufsm.br,batista@inf.ufsm.br

Grupo de Microeletrônica - GMICRO
Universidade Federal de Santa Maria - UFSM

Abstract

This paper presents enhancements based on a framework for Wireless Sensor Networks (WSNs) based on Matlab tools (Simulink, Stateflow and Stateflow Coder). The application developer can model the network using Stateflow constructs, in a single model to perform multi-platform Hardware-in-The-Loop (HIL) simulation. Constrains in WSN, like energy limitation, range and fault tolerance can be a problem to deploying a large number of sensors in the real world under different physical scenarios. A simulation environment is feasible choice to build prototype models for simulating the nodes and network behaviors. The design of a reliable framework is a big challenge and a hot research topic in WSN. In order to validate this framework, a localization algorithm was designed. Resulting from the algorithm implementation, the framework has undergone some changes, such as, removing the Super Sensor Block, connection table based in distance between nodes, automation of some functions of the framework configuration and mobility of nodes.

1. Introduction

Interconnect the physical world with the virtual one often provides opportunities to explore fast solutions. It is even more relevant for WSNs applications, where it is common for applications developed that need to be tested and implemented in dozens, hundreds or even thousands of nodes. However, it is impractical to manage real tests in a sensor network in its development phase, or when there are a very large number of nodes. One solution is to rely on a framework*[1]*for simulation that allows developers to create virtual nodes and provide a certain level of abstraction, able to implement different applications on different nodes.

Wireless Sensor Networks have become a promising topic of current and future microelectronic technologies. Currently,is being developedsome different types of software applications for WSNs, the implementationof these applications is usually performed using aplatform, such as TinyOS[2], MANTIS [3], FreeRTOS [4] etc.Verify these applications is challenging,most of the available sensor nodes (e.g. MicaZ [5] andTmote Sky [6]) provide a few on-board blinking ledsas debug. Because of this, encoding on theseplatforms is very difficult *[7]*.

In order to outline these difficulties, it is needed to be ableto construct the application using high level abstractions, such as Stateflow state charts, andsimulate it using configurable and realistic topologies for thenetwork itself.

The developer can configure the connectivity of the nodes based on its positions and can perform the simulationand verification of the application. The developer can also use the broad variety tools provided by MathWorks*[8]*, suchas animated state chart displays, plots, scopes and a large number of available Simulinkblocks.

In the following, its being modeled a very simple connectivity metric method for localization *[9]* that makes use of the radio-frequency (RF) communications capabilities. By doing this, the effectiveness of the present framework is validated.

The paper is organized as follows. Next we introduce the global architecture of the framework. Section 3 describes the framework validation through a localization algorithm implementation and simulation. Finally, we conclude the paper in Section 4.

2. Global architecture of the framework

A block diagram level view of this simulation tool is shown in fig.1.

A description of the functionality of the components in the framework (global data structure, system clock block, nodes block and super-medium block) is given in details in this section.

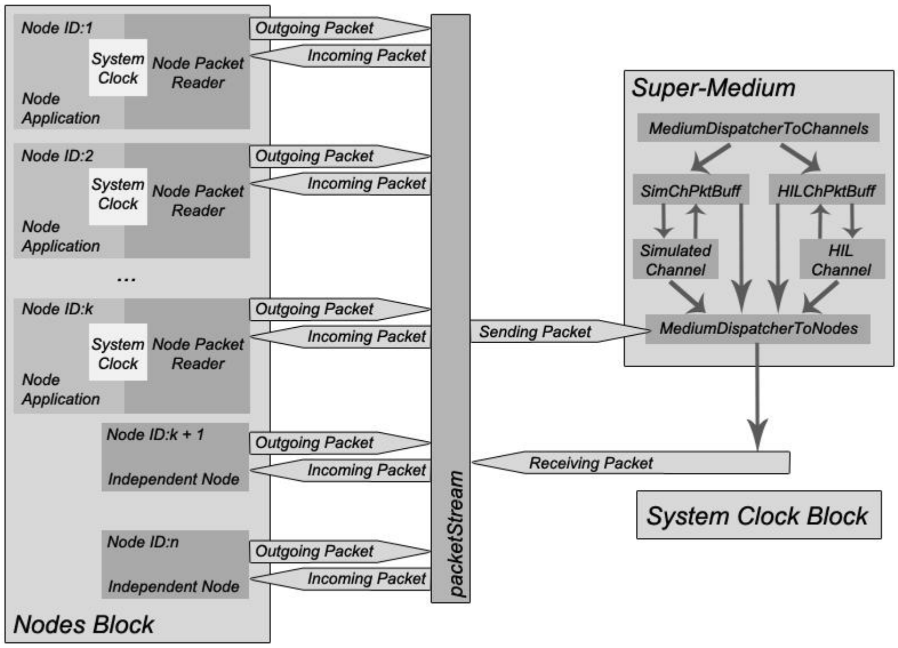

Fig. 1 – Logical components

2.1. Global data structure

It is used for configuring and managing the simulation framework and it is composed of data structures (i.e. nbrNodes, packetStream, ndType and soon).

nbrNodes:is a scalar which stores the number of nodes in the framework, includes the fully simulated nodes(SIM), partially simulated nodes (HIL_SEN, HIL_RF, HIL_FULL) and independent nodes (REAL).

packetStream:provide the bridge for communication between Nodes Block andSuper-Medium Block.Store the packets that the nodes are going to transmit to the Super-Medium Block, the packetsthat the nodes are going toreceive from Super-Medium Block and the corresponding valid bits of the packets for the nodes in the framework.

ndType:is a vector which defines the role of each node in the framework. Anode can be one of the following types: fullysimulated (SIM), partially simulated (HIL_SEN, HIL_RF, HIL_FULL) and independent (REAL).

2.2. System clock block

Provide the clock signal to all the blocks(global data structure, system clock block, nodes block and super-medium block) for implementingthe global synchronization in the framework.

2.3. Nodes block

Consist of the nodes simulated in the framework, SIM nodes, HIL_SEN nodes, HIL_RF nodes, HIL_FULL nodes and REAL nodes. It is constructed by three logical segments: System Clock (receives the synchronous signal), Node Packet Reader (listens to the Super-Medium Block for detecting if there is a received packet to the node) and Node Application (algorithm executed by the node).

2.4. Super medium block

Manage the packet transmission in the framework between simulated nodes, and the transmissionbetween simulated nodes and independent nodes. It is composed of System Clock (synchronous signal), Medium_DispatcherToChannels (transmitting packet mechanism that delivers the packet from the node side to the channel side),Simulated_Channel (performs the channel behavior on the packets transmittingthrough it in a virtual mode), HIL_Channel(performs the channel behavior on the packets transmittingthrough it in a real mode) and Medium_DispatcherToNodes (implements the receiving packet mechanism by meansof delivering the packet from the channel side to the node side).

3. Framework validation

Thissectionwill show the framework validation through the implementation of a localization algorithm. The simulation was performed considering the mobility of nodes (190 positions in a field of 20 meters long), but, for the purpose of this paper, it will be show the result of just one position.

3.1. The localization algorithm

For the localization algorithm some assumptions were made, such as, perfect spherical radio propagation and identical transmission range for all nodes.

For this simulation, the network consists of 10 (ten) nodes, which 9 (nine) serve as reference point and 1 (one) is the lost node. All the nodes have known positions (X_1,Y_1) to (X_{10},Y_{10}) shown in tab.1 that transmit their respective beacon signal containing their respective IDs and positions, the lost node collects all the reference nodes positions that heis connected based in a connection table, where CON means connected and NC not connected.Tab. 2is filled according to an arbitrary distance (15 meters for this simulation) between the respective node and the lost node.

Tab.1 – Nodesposition

NodeID	1	2	3	4	5	6	7	8	9	10
X Pos.(1)	0.00	0.00	20.00	20.00	7.00	7.00	14.00	14.00	15.70	6.20
YPos.(1)	0.00	20.00	0.00	20.00	7.00	14.00	7.00	14.00	12.70	1.10

Tab.2 – Connection tablebetween the nodes

NodeID	1	2	3	4	5	6	7	8	9	10
1	NC	NC	NC	NC	CON	NC	NC	NC	NC	CON
2	NC	NC	NC	NC	CON	CON	NC	NC	NC	NC
3	NC	NC	NC	NC	CON	NC	CON	NC	CON	CON
4	NC	NC	NC	NC	NC	CON	CON	CON	CON	NC
5	CON	CON	CON	NC	NC	CON	CON	CON	CON	CON
6	NC	NC	NC	CON	CON	NC	CON	CON	CON	CON
7	NC	NC	CON	CON	CON	CON	NC	CON	CON	CON
8	NC	NC	NC	CON	CON	CON	CON	NC	CON	NC
9	NC	NC	CON	CON	CON	CON	CON	CON	NC	CON
10	CON	NC	CON	NC	CON	CON	CON	NC	CON	NC

Fig.2 show the two dimension graphical view (first of 190 available positions) of tab.1 and tab.2 by means of which nodes are connected (nodes 1, 3, 5, 6, 7 and 9) with the lost node. Node number 10 represented by a circular marker is the lost node with his know position (6.2,1.1).The approximated location of the lost node is placed in (10,6) represented by a X marker.

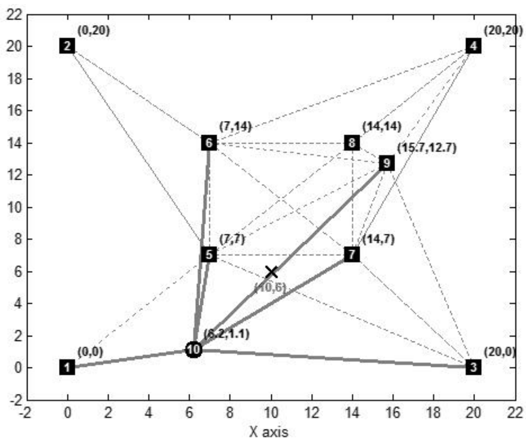

Fig.2 – Two dimension view of reference nodes connectivity

For the localization algorithm calculation, the following mathematical formulas were used:

$$CM_i = \frac{N_{recv}(i)}{N_{sent}(i)} X\ 100 \quad (1)$$

Where:

CM is the connectivity metric for the respective reference node

$N_{sent}(i)$is the number of beacons that have been sent by the respective reference node

$N_{recv}(i)$is the number of beacons that have been received by the lost node from the respective reference node.

The lost node will use for the calculation the respective reference node if $CM_{Threshold} = 90\%$.

To perform the localization calculation of the lost node, the following formula is used:

$$(X_{est}, Y_{est}) = \left(\frac{X_1 + \cdots + X_n}{K}, \frac{Y_1 + \cdots + Y_n}{K}\right) \quad (2)$$

Where:

$(X_1 + \cdots + X_n)$ is the sum of X positions from the reference nodes.

$(Y_1 + \cdots + Y_n)$ is the sum of Y positions from the reference nodes.

K is the number of reference nodes used for the calculation.

E.g.

$$(X_{est}, Y_{est}) = \left(\frac{0.0 + 20.0 + 7.0 + 7.0 + 14.0 + 15.7}{6}, \frac{0.0 + 0.0 + 7.0 + 14.0 + 7.0 + 12.7}{6}\right)$$

Therefore,

$$(X_{est}, Y_{est}) = (10.62, 6.78) \text{against} (X_{10}, Y_{10}) = (6.20, 1.1)$$

To characterize the accuracy of the estimate, the localization error (LE) is defined:

$$LE = \sqrt{(X_{est} - X_a)^2 + (Y_{est} - Y_a)^2} \quad (3)$$

Where:

$X_a$andY_a are the actual location of the lost node.

$$LE = \sqrt{(6.20 - 10.62)^2 + (1.1 - 6.78)^2} = 7.20$$

If the number of reference nodes is increased, it is know that the accuracy of the localization estimate improves.

4. Conclusion

In this paper, has beenpresented a framework for Wireless Sensor Networks based on Matlab tools, also an implementation to calculate the estimated position of a node in a network was performed. By performing the simulation shown in section 3, obtaining the correct results and making possible to perform the mobility support, the framework was validated for its purpose.

Based on simulation services offered by this framework, developers can model a WSN and make a simulation code that can be implemented with a high level of abstraction before the implementation in the node. They can also pre-analyze and verify the implementation of the application platform and operation system independent. This has the advantage of reducing the cycle time for the design and cost, also improve the portability of an application for WSN.

5. References

[1] M. M. R. Mozumdar, L. Lavagno, L. Vanzago and et. al. “A Framework for Modeling, Simulation and Automatic Code Generation of Sensor Network Application”, In the Proceedings of the Fifth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks -SECON, San Francisco, June 2008.

[2] TinyOS Community Forum, An open-source OS for the networkedsensor regime. http://www.tinyos.net.

[3] MANTIS, MultimodAlNeTwork of In-situ Sensors.http://mantis.cs.colorado.edu/index.php/tiki-index.php.

[4] FreeRTOS, A FREE open source RTOS for small embedded real time systems. http://www.freertos.org/PC/

[5] Crossbow Technology: Inertial Systems. http://www.xbow.com.

[6] Sentilla, Pervasive Computing Solutions. www.sentilla.com.

[7] Esteban Egea-Lopez, Javier Valer-Alonso, Alejandro Martinez-Sala, Pablo Pavon-Marino and Joan Garicia-Haro, “Simulation scalability issues in wirelesssensor networks”, Communications Magazine, IEEE, Vol. 44, Issue 7, July 2006. Page(s): 64-73.

[8] MATHWORKS.“MATLAB and Simulink for Technical Computing”.http://www.mathworks.com.

[9] Heidemann, J., Bulusu, N. and Estrin, D.,“GPS-less Low Cost Outdoor Localization ForVery Small Devices”. IEEE Personal Communications Magazine, Vol. 7, No. 5, pp. 28-34. October, 2000.