A Delay-Sensitive Mathematical Model Approach and a Distributed Algorithm for Mobile Wireless Sensor Networks

Germán A. Montoya
System and Computing Engineering Department
Universidad de Los Andes
Bogotá, Colombia
ga.montoya44@uniandes.edu.co

Yezid Donoso
System and Computing Engineering Department
Universidad de Los Andes
Bogotá, Colombia
ydonoso@uniandes.edu.co

***Abstract*—Communication disruptions caused by mobility in wireless sensor networks introduce undesired delays which affect the network performance in delay sensitive applications in MWSN. In order to study the negative effects caused by mobility, we propose a mathematical model to find the minimum cost path between a source node and a destination node considering the nodes position changes across time. Our mathematical model considers the usage of buffers in the nodes to represent the fact of storing a message if there is not an appropriate forwarding node for transmitting it. To contrast our mathematical model results we have designed two kinds of algorithms: the first one takes advantage of the closest neighbours to the destination node in order to reach it as fast as possible from the source node. The second one simply reaches the destination node if a neighbour node is precisely the destination node. Finally, we compare the delay performance of these algorithms against the mathematical model to demonstrate how efficient they are for reaching a destination node.**

Index Terms-**Mathematical model; Delays; MWSN;**

I. INTRODUCTION

The advances of Wireless Sensor Networks (WSN) have allowed attaching the sensors to entities such as objects, animals or humans, for monitoring a physical variable presented in a particular environment. However, these sensors are equipped with limited batteries whereby it is required to implement energy efficient routing techniques to extend as much as possible the lifetime of these devices [1][2]. Moreover, communication disruptions caused by mobility in wireless sensor networks introduce undesired delays which affect the network performance in delay sensitive applications, such as military or healthcare monitoring applications. For the latter, due to they deal with health states, illness and continuous medical supervision, a base station should no experiment delays from the information collected by the sensors [3][4].

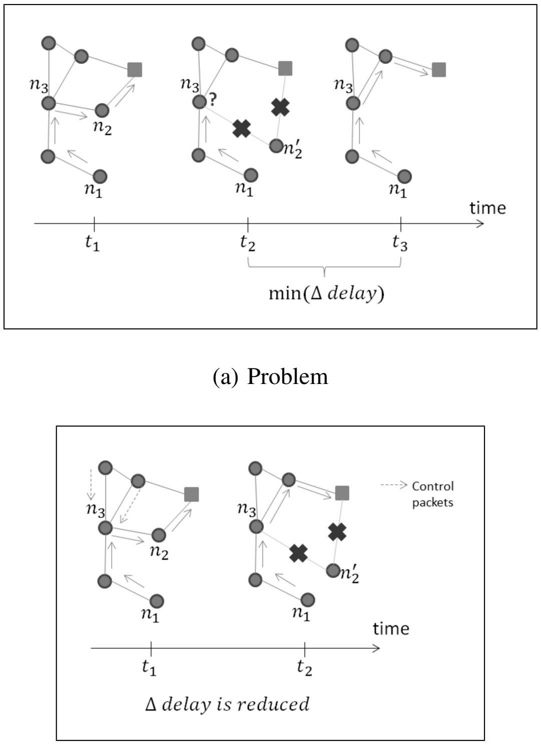
Given the scenario described above, novel routing algorithms are emerging for solving these delays problems [7] [8] [9]. However, these algorithms need to be compared against a mathematical model formulation and formulation which can give us an optimal solution in order to do valuable comparisons. To study the negative effects caused by mobility, we propose a mathematical model to find the minimum cost path between a source node and a destination node considering the nodes position changes across time. Our mathematical model considers the usage of buffers in the nodes to represent the fact of storing a message if there is not an appropriate forwarding node for transmitting it.

In order to contrast our mathematical model results we have designed two kinds of algorithms: the first one takes advantage of the closest neighbours to the destination node in order to reach it as fast as possible from the source node. The second one simply reaches the destination node if a neighbour node is precisely the destination node.

The remainder of the paper is organized as follows: Section II describes the problem statement in a general view. Section III presents the problem formulation, that is, how the problem is described as a modular problem. Section IV shows the mathematical model proposed, the objective function and the constraints. Section V presents the proposed algorithms for comparing them against the optimal solution given by the mathematical optimization model. Finally, the sections VI and VII show the results and conclusions respectively.

II. GENERAL PROBLEM STATEMENT

The figure 1.a) presents the problem we want to solve. Suppose we have a MWSN where at time t_1 there is a communication path between the source sensor node n_1 and the base station. However, at time t_2, the node n_2 moves away from the node n_3, causing a communication disruption for transmitting the information from n_1 to the base station or destination node. Once n_3 has realized of this problem at time t_3, n_3 has to perform routing corrections in order to reestablish the communication path between n_1 and the base station. The communication reestablishment between n_1 and the base station can be perfectly performed using routing techniques, but at the expense of introduce undesired delays for building again the communication path between n_1 and the base station. In some applications these delays can be omitted because do not affect the purpose itself of the application, but in other ones, such as delay sensitive applications like military



(a) Problem

(b) Solution

Fig. 1: Problem Definition

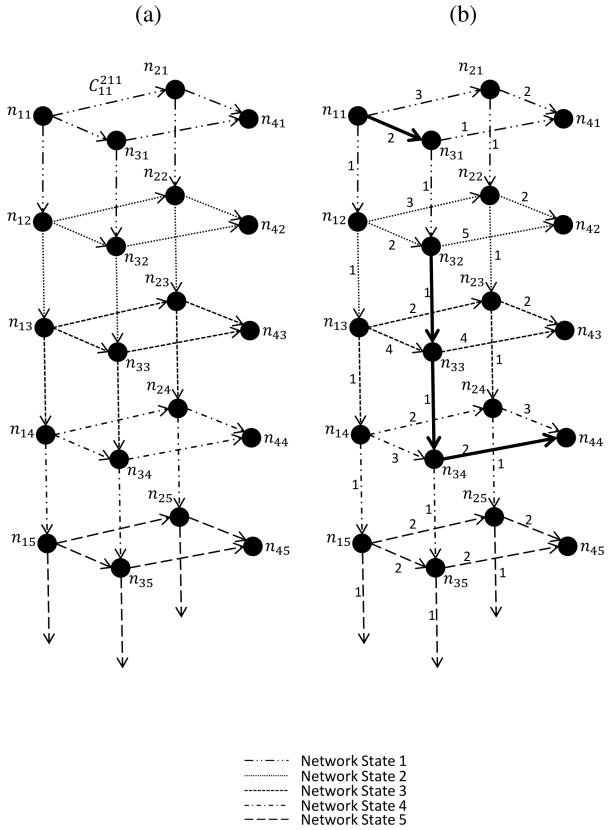


Fig. 2: Problem Scenario.

or health applications, this disadvantage might mean a very low network performance in terms of delays.

Given the problem above, our proposal consists to design a mathematical model for finding the minimum cost path between a source node and a destination node taking into account that the network is changing across time [5] [6]. This formulation would be very important because it can give us optimal values for contrasting with algorithms results.

III. Problem Formulation

In this section our problem is enunciated and described in detail, as well as some assumptions are shown in order to simplify our proposed mathematical model.

Based on the figure above, we will describe our problem:

- Mobile Network: Assume we have a mobile network, at which the nodes position changes across time periods. Due to the nodes position changes, the links cost between the network nodes also changes across time periods. This means that at each time period the network has particular links cost, different from the links cost at other time period. Given that at each time period the network have different links cost, we could say this reflects the network state in a given time period. For this reason, each network at a given time period will be called *Network State*. For instance, the *Network State* at time period *1* is called *Network State 1*, the *Network State* at time period *2* is called *Network State 2*, and so on. In other words, according to the figure 2.a) we have an initial network (Network State 1) compound by 4 nodes. As these nodes conform a network, there are interrelations between them that we will call *Links*. These links can have a cost, which can be represented, for example, by the distance, the signal to noise ratio or the RSSI measurement between the nodes. In the next time period, the network costs at the Network State 1 change and then these new interrelations between the nodes are now the *Network State 2*. As the next time period occurs, the network at the Network State 2 becomes the network at the Network State 3, and this network will be the network at the network State 4, and so on.
- Nodes: Each node is denoted as n_{it} where i is number of the node and t is the network state of the node. Depending on the communication range, a node can communicate with another node in the direction described by the figure. For example, n_{11} can communicate with n_{21} and n_{31}.
- Buffer in each node: In telecommunication networks, a router or a sensor (a node) can decide not-sending its message, storing it in a buffer until it would be appropriated to send it. In our model, this situation is represented as a link between n_{11} and n_{12}, meaning that n_{11} can store its message in its buffer, that is, the node n_{12}.

- Costs: As it was mentioned previously, a link has a cost. Then, there is a cost for sending a message from n_{11} to n_{21} called C_{11}^{21l}, and denoted as C_{it}^{jul}, that is, this is the cost to carry a message form the node i at the state t to the node j at the state u at the Network State l.
- Directed graph: In this example our goal consists to carry a message from the node 1 to the node 4. Then, our *Source* node is the node 1, and our *Destination* node is the node 4. In this sense, a directed graph is constructed from the Source to the Destination. For this reason, the links direction points to the Destination.
- Goal: Our goal consists to carry a message from the Source node to the Destination node using the neighbours nodes as forwarding nodes for passing the message, and even using the buffers, if it is necessary, for waiting an appropriated situation for sending the message. In this sense, we have to find the minimum cost path between a Source node and a Destination node considering the network is changing across time, that is, through the Network States. Additionally, for simplicity we assume only one link can be selected for sending the message per each Network State. This means that if a message is at the node n_{11}, this node at this Network State 1 can send the message to only one neighbour, n_{21} or n_{11}, or storing it in its buffer, that is, n_{12}.
- Example Result: According to the example shown in the figure 2.b) and based on the links cost, the minimum cost path from the Source node, n_{11}, to the Destination node, node 4, is the path compounded by the highlighted links: n_{11} to n_{31}, n_{32} to n_{33}, n_{33} to n_{34} and n_{34} to n_{44}. As we will describe later in the mathematical formulation, this result also can be expressed in terms of X: $X_{11}^{311} = 1$, $X_{32}^{332} = 1$, $X_{33}^{343} = 1$ and $X_{34}^{444} = 1$. The rest of X_{it}^{jul} values are zero.

IV. Mathematical Model Solution

In this section is proposed a mathematical model for finding the minimum cost path between a source node and a destination node considering the network is changing across time. The next table presents the sets, parameters and decision variables for building the mathematical model.

Minimize:

$$min\sum_{i\in N}\sum_{t\in S}\sum_{j\in N}\sum_{u\in S}\sum_{l\in S}C_{it}^{jul}X_{it}^{jul} \tag{1}$$

Subject to:

$$\sum_{j\in N}\sum_{u\in S}X_{it}^{jul}Y_{jl}=1$$
$$\forall i\in N \mid i=o, \quad \forall t,l\in S \mid t=1,l=1 \tag{2}$$

$$\sum_{i\in N}\sum_{t\in S}\sum_{j\in N}\sum_{u\in S}X_{it}^{jul}Y_{im}Y_{jl}=1$$
$$\forall l,m\in S \mid l<st, m=l-1 \tag{3}$$

TABLE I: Sets, Parameteres and Variables description

Sets	**Description**
N	Nodes set.
S	States set.
Parameters	**Description**
o	Source node.
d	Destination node.
st	State at which we want to obtain the minimum cost path from the *Source* to the *Destination*.
C_{it}^{jul}	Link cost from the node *i* at the state *t* to the node *j* at the state *u* at the network state *l*.
Variables	**Description**
X_{it}^{jul}	Determines if the link at the state *l* from the node *i* at the state *t* to the node *j* at the state *u* is selected for building the path towards the *Destination* (Binary variable).
$Y_{i,l}$	Determines if the node *i* at the state *l* is selected as a forwarding node for building the path towards the *Destination* (Binary variable).

$$\sum_{j\in N}Y_{jl}=1 \qquad \forall l\in S \mid l\leq st \tag{4}$$

$$\sum_{i\in N}\sum_{t\in S}X_{it}^{jul}Y_{im}Y_{jl}=1$$
$$\forall j\in N \mid j=d$$
$$\forall u,l,m\in S \mid l<st, u=l=st, m=l-1 \tag{5}$$

The equation 1 corresponds to the objective function, which attempts to find the X_{it}^{jul} variables at the minimum cost. The expression 2 establishes that once a X_{it}^{jul} is selected considering that $i=o$, $t=st1$ and $l=NetworkState1$, we can know the forwarding node j for sending the message. The equation 5 determines the predecessor node i at the network state m (previous to the network state l) required for building the path. The expression 3 allows to be coherent the forwarding and predecessor nodes in the intermediate states, that is, the network states different from the Network State 1 and the Network State indicated by the parameter st. Finally, the equation 4 assures that only one X_{it}^{jul} must be selected at each Network State from the Network State 1 up to the Network State indicated by the parameter st.

In summary, this mathematical model gives us the minimum cost path by introducing the following parameters: the Source node, the Destination node and the State (the Network State) at which we want to obtain the minimum cost path from the Source node to the Destination node. However, in this model we cannot prescind from the State parameter, that is, we cannot know the Network State (from all the Network States) at which we would obtain the minimum path cost. For this reason, it is required to use this mathematical model in an iterative way, that is, obtaining the minimum path cost for all the Network State and then, selecting the lowest of the minimum paths obtained. This process is represented in the Algorithm 1.

In this algorithm, the mathematical model is used for each Network State, and once we have analysed all the Network States, we finally determine which one obtained the minimum cost path , denoted as $minSolution$, at which Network State,

Algorithm 1 Solution Pseudocode

```
$o = source;\ d = destination;\ k = number\ of\ states$
$minLocal = \infty; minState = 0$
for $i = 1$ to $k$ do
    $minSolution = MathModel(o,\ d,\ i)$
    if $minSolution < minLocal$ then
        $minLocal = minSolution$
        $minState = i$
    end if
end for
```

expressed as $minState$. In summary, with the mathematical model and this algorithm we can obtain the minimum cost path given a Source node and a Destination node.

V. Algorithms Proposals

To contrast the mathematical model results we have designed two kinds of algorithms, which will be described below:

A. Single Path with Connection to the Sink (SIPCOS)

As we saw previously in the mathematical model, our goal consists to find the minimum cost path between a Source node and a Destination node. We are going to assume that a link cost represents the delay for sending the information from a node to another one. For simplicity, we assume that each link cost (delay) will be proportional to the distance between two nodes in the network. Therefore, our algorithm will try to reach the Destination node at the minimum delay possible from the Source node. In the next figure is presented a network example.

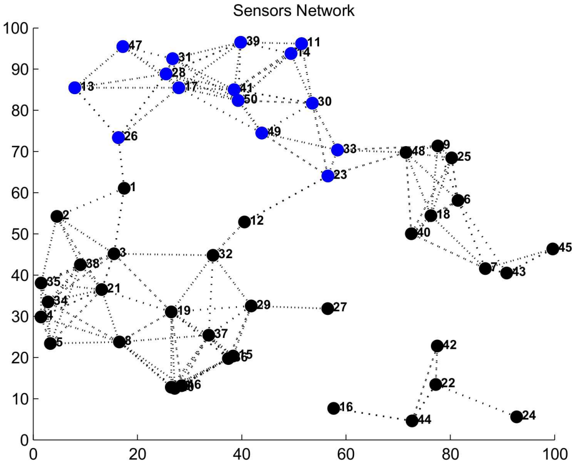


Fig. 3: Problem Scenario.

According to the Figure 3, suppose the source node is the node 3, and the Destination node is the node 50. Our proposal includes two methods. The first one consists to perform a partial broadcast from the Destination node to the closest neighbours. Now, these neighbours, the blue ones (called *connected nodes to the destination*) know how to reach quickly the Destination node in such case that a message arrives to them. The second method consists to send a message from the Source node to forwarding nodes in order to reach a connected node to the Destination. This method intends to reach as fast as possible the destination node, trying to obtain similar values respect to the mathematical model values. The pseudocode of these algorithms is described below:

Algorithm 2 Destination Node Algorithm

```
$d = destination$
$r = number\ of\ rounds$
$dn = neighbours(d)$                          ▷ $dn$: destination neighbours
while $r > 0$ do
    $dn = neighbours(dn)$
    $sendControlMessage(dn)$
    $r = r - 1$
end while
```

Algorithm 3 Forwarding Node Algorithm

```
parameter $s$                                  ▷ Source Node
parameter $d$                                  ▷ Destination Node
parameter $dN$                                 ▷ Destination Neighbours Nodes
array $path = []$                              ▷ The Building Communication Path
variable $fn = s$                              ▷ $fn$: Forwarding Node
function $FN(i)$                               ▷ $FN$: Forwarding Neighbours Function
$minCost = \infty$
$DestinationNeighbourFound = 0$
array $path = [path\ \ fn]$
while $message\ \notin\ dN\ \vee\ d$ do
    $forwardingNeighbours = neighbours(fn)$
    for $i = 1\ to\ forwardingNeighbours$ do
        if $FN(i)\ \in\ dN$ then
            $DestinationNeighbourFound = 1$
            $sendMessage(FN(i))$
            $path = [path\ \ FN(i)]$
        end if
    end for
    if $DestinationNeighbourFound == 0$ then
        for $i = 1\ to\ forwardingNeighbours$ do
            $neighbourCost = cost(fn,\ FN(i))$
            if $neighbourCost < minCost$ then
                $fn = FN(i)$
                $sendMessage(fn)$
                $path = [path\ \ fn]$
            end if
        end for
    end if
end while
if $message\ \in\ dN$ then
    $dN = neighbours(dN)$
    if $Destination\ \in\ dN$ then
        $sendMessage(d)$
        $path = [path\ \ d]$
    else
        $sendMessage(dN)$
        $path = [path\ \ dN]$
    end if
end if
```

Next, the pseudocodes are briefly explained:

- Destination Node Algorithm: this algorithm defines a simple technique to select the neighbours nodes connected to the Destination node. This technique consists to send from the Destination node a control message to the Destination neighbours, and these ones send this control messages to its neighbours many times as the parameter *round* allows it.

- Forwarding Node Algorithm: this algorithm is initially performed in the Source node. Therefore, this node selects a neighbour node based on its low cost to transmit the message in order to reach a connected node to the Destination or the Destination itself. Once a neighbour node is selected for sending the message, this node is added to the array $path$. Finally, when the Destination is reached, the array $path$ contains the message's traceability from the Source node to the Destination node.

B. Single Path without Connection to the Sink (SIP)

In contrast to the SIPCOS algorithm, the SIP algorithm does not take into account the connected nodes to the Destination. For this reason, the algorithm must find exactly the Destination. Therefore, it is too much difficult to find the Destination using this method. The pseudocode of this algorithm is described in the Algorithm 4.

Algorithm 4 Forwarding Node Algorithm

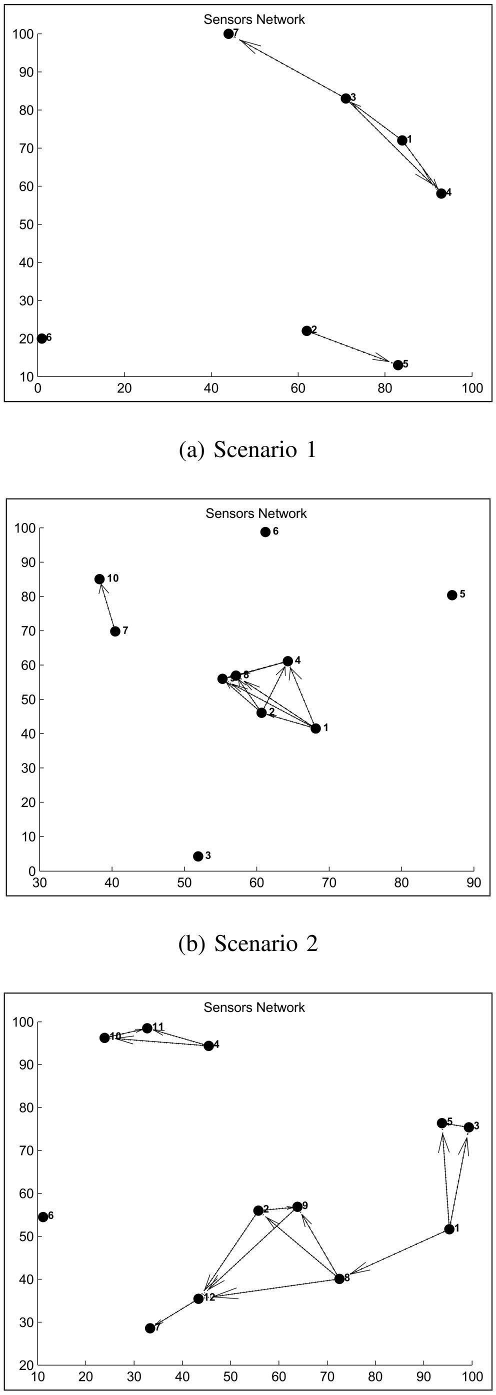
```
parameter $s$
parameter $d$
$array\ path = []$
variable $fn = s$
function $FN(i)$
$minCost = \infty$
array $path = [path\ \ fn]$
$DestinationFound = 0$
while $message\ \notin\ d$ do
    $forwardingNeighbours = neighbours(fn)$
    for $i = 1\ to\ forwardingNeighbours$ do
        if $forwardingNeighbours(i)\ ==\ d$ then
            $DestinationFound = 1$
            $sendMessage(d)$
            $path = [path\ \ FN(i)]$
        end if
    end for
    if $DestinationFound == 0$ then
        for $i = 1\ to\ forwardingNeighbours$ do
            $neighbourCost = cost(forwardingNode,\ FN(i))$
            if $neighbourCost < minCost$ then
                $forwardingNode = FN(i)$
                $sendMessage(fn)$
                $path = [path\ \ fn]$
            end if
        end for
    end if
end while
if $message\ \in\ d$ then
    $path = [path\ \ d]$
end if
```

VI. Results

The graphs for the three scenarios, at its first Network State, are shown in the Figure 4.

There were chosen 3 basic scenarios in order to test in a basic way the behavior of the mathematical model, the SIPCOS and SIP algorithms. The results are shown in the Tables II, III and IV.

According to the first scenario, the SIPCOS algorithm showed the same delay respect to the optimal value, while the SIP algorithm used an extra delay to reach the destination. This extra delay is a low value because this scenario corresponds to



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Fig. 4: Network Scenarios

TABLE II: Results for the Mathematical Model

Parameters and Variables	Scenario 1	Scenario 2	Scenario 3
Nodes	7	10	12
States	14	20	24
Source node	1	1	1
Destination node	7	10	12
Solution Path	1,3,7	1,2,8,7,10	1,8,12
Delay	2	4	2

TABLE III: Results for SIPCOS Algorithm

Parameters and Variables	Scenario 1	Scenario 2	Scenario 3
Nodes	7	10	12
States	14	20	24
Source node	1	1	1
Destination node	7	10	12
Solution Path	1,3,7	1 2 9 7 10	1,8,2,12
Delay	2	4	3

TABLE IV: Results for SIP Algorithm

Parameters and Variables	Scenario 1	Scenario 2	Scenario 3
Nodes	7	10	12
States	14	20	24
Source node	1	1	1
Destination node	7	10	12
Solution Path	1,4,3,7	1,2,8,9,7,10	1,3,5,9,2,12
Delay	3	5	5

a small network. For this reason, if a network has few nodes, there is more probable the SIP algorithm reach the SIPCOS's performance. Once the network has increasing in size, such in case the scenario 2 or 3, the SIP's performance decreases. Finally, for the third scenario the SIPCOS algorithm presented a better performance respect to the SIP algorithm, thanks to the connected nodes which allowed reach the destination node as fast as possible. In this scenario the SIP algorithm lose too much time in the nodes 3 and 5, because it had not a clear strategy to go out from ending nodes.

VII. Conclusions

In this paper we propose a mathematical model which is able to find the minimum cost path between a source node and a destination node considering a mobile network. Additionally, there were proposed two algorithms, the SIPCOS and SIP algorithms, in order to compare their results against the mathematical model. The results showed the SIPCOS has a better performance compared against the SIP algorithm because it has efficient strategies for find quickly the destination node or destination neighbours, which facilitates the fact of building and finding the communication path from the source node to the destination node. For this reason, the SIPCOS's performance was very similar to the mathematical model and it could be a good option to be used in MWSN.

References

[1] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, 2010.

[2] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*, 2009.

[3] A. A. Ahmed, *An enhanced real-time routing protocol with load distribution for mobile wireless sensor networks*. Computer Networks, vol.57, 2013.

[4] A. A. Ahmed, *Real-Time Wireless Sensor Networks*. University of Virginia, 2007.

[5] B. Buchli, F. Sutton and J. Beutel. *GPS-Equipped Wireless Sensor Network Node for High-Accuracy Positioning Applications*. Wireless Sensor Networks Lecture Notes in Computer Science, Vol. 7158, pp 179-195, Springer, 2012

[6] S. Li, X. Ma, X. Wang, M. Tan. *Energy-efficient multipath routing in wireless sensor network considering wireless interference*. Journal of Control Theory and Applications. Vol. 9, Issue 1, pp. 127-132. 2011.

[7] G. M. de Araújo, J. Kaiser and L. B.Becker . G*enetic Machine Learning Approach for Link Quality Prediction in Mobile Wireless Sensor Networks*. Cooperative Robots and Sensor Networks. 2014.

[8] G. M. de Araújo, J. Kaiser and L. B.Becker. *An Optimized Markov Model to Predict Link Quality in Mobile Wireless Sensor Networks*. 2012.

[9] J. A. Torkestani Young. *Mobility prediction in mobile wireless networks*. Journal of Network and Computer Applications. Vol.35, 2012.