

# Delay-Sensitive Optimization Models and Distributed Routing Algorithms for Mobile Wireless Sensor Networks

G.A. Montoya, Y. Donoso

**Germán A. Montoya, Yezid Donoso\***

System and Computing Engineering Department  
Universidad de Los Andes  
Bogotá, Colombia, South America  
ga.montoya44@uniandes.edu.co

\*Corresponding author: 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 two mathematical models to find the minimum cost path between a source node and a destination node considering the nodes position changes across time. Our mathematical models consider 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 models 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 our mathematical models to show how efficient they are for reaching a destination node. This paper is an extension of [10].<sup>a</sup> The mathematical model proposed in [10] is improved by adding two new binary variables with the aim of make it more readable and compact mathematically. This means a post-processing algorithm is added only for evaluating if a solution is at the first network state.

**Keywords:** Mathematical model, Delays, MWSN.

---

<sup>a</sup>Reprinted (partial) and extended, with permission based on License Number 3961371369962 ©[2016] IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on".

## 1 Introduction

The advances of Wireless Sensor Networks (WSN) have allowed attaching sensor devices 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]. Moreover, these algorithms need to be compared against a mathematical model for knowing their delay performance in terms of delay. With the purpose of analyse 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. This paper is an extension of [10] (doi: 10.1109/ICCCC.2016.7496736). The mathematical model proposed in [10] is improved by adding two new binary variables with the aim of make it more readable and compact mathematically. This means a post-processing algorithm is added only for evaluating if a solution is at the first network state.

In order to contrast our mathematical model results we have designed two 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 2 describes the problem statement in a general view. Section 3 presents the problem formulation, that is, how the problem is described as a modular problem. Section 4 shows the mathematical model proposed, the objective function and the constraints. Section 5 presents the proposed algorithms for comparing them against the optimal solution given by the mathematical optimization model. Finally, sections 6 and 6 show the results and conclusions respectively.

## 2 General problem statement

The Figure 1.a presents the problem we want to solve, where a rounded node is a sensor and a square node corresponds to a base station (destination node). 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 information from  $n_1$  to the base station. 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. This reestablishment can be perfectly performed using routing techniques, but at the expense of introducing 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 application purpose, but in other cases, such as delay sensitive applications like military or health applications, this disadvantage might mean a very low network performance in terms of delay.

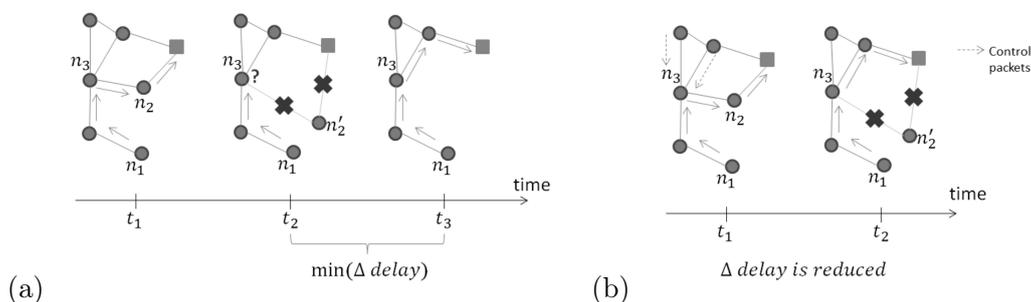


Figure 1: Problem definition: (a) Problem; (b) Solution

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 considering the network is moving across time [5] [6]. This formulation would be very important because it can give us optimal values for contrasting with algorithms results.

### 3 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 models.

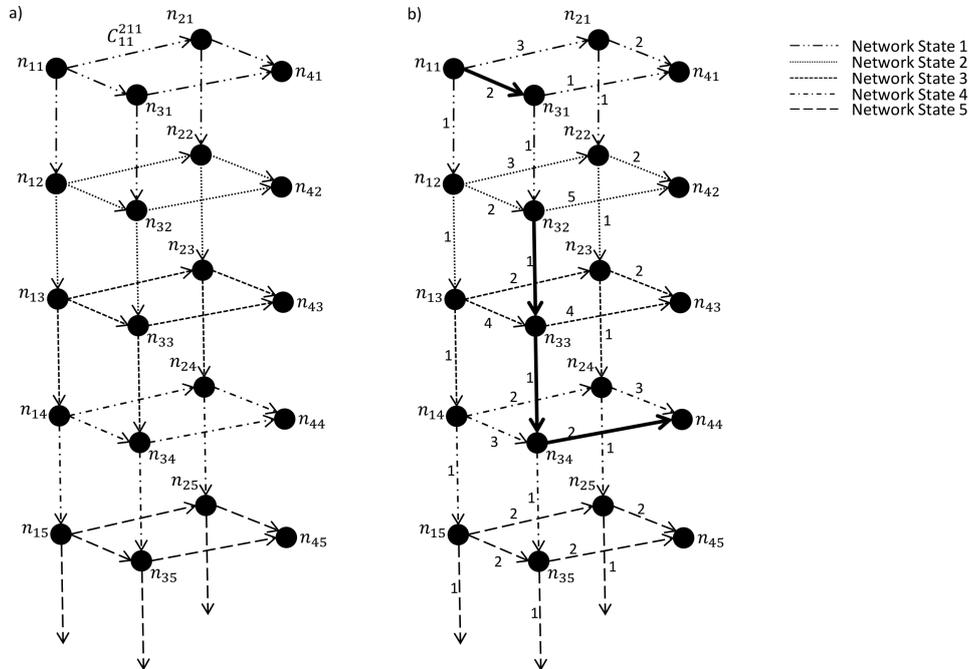


Figure 2: Problem scenario

Based on the Figure 2, we will describe our problem:

- **Mobile Network:** Assume we have a mobile network, at which the nodes position changes across time periods. For this reason, 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 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 2. 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 a 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 a 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 can be also 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.

## 4 Mathematical model solution

Two mathematical models are proposed, which achieve same optimal solutions since they are equivalents, but the second one is mathematically more complete than the first one.

### 4.1 First approach

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 Table 1 presents the sets, parameters and decision variables for building the mathematical model.

Next, our mathematical model is enunciated.

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

Subject to:

Table 1: Sets, parameters and variables description

Sets and Parameters	Description
$N$	Nodes set.
$S$	States set.
$o$	Source node.
$d$	Destination node.
$st$	State at which we want to obtain the minimum cost path from the <i>Source</i> to the <i>Destination</i> .
$C_{it}^{jul}$	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 <i>Destination</i> (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 <i>Destination</i> (Binary variable).

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

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

$$\sum_{j \in N} Y_{jl} = 1 \quad \forall l \in S \mid l \leq st \quad (4)$$

$$\sum_{i \in N} \sum_{t \in S} X_{it}^{jul} Y_{im} Y_{jl} = 1 \quad (5)$$

$$\forall j \in N \mid j = d, \quad \forall u, l, m \in S \mid l < st, u = l = st, m = l - 1$$

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 4.1 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.

---

**Algorithm 1** Solution pseudocode
 

---

```

1:  $o = source; d = destination; k = number\ of\ states$ 
2:  $minLocal = \infty; minState = 0$ 
3: for  $i = 1$  to  $k$  do
4:    $minSolution = MathModel(o, d, i)$ 
5:   if  $minSolution < minLocal$  then
6:      $minLocal = minSolution$ 
7:      $minState = i$ 
8:   end if
9: end for

```

---

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, 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.

## 4.2 Second approach

In this section is presented a second mathematical model proposed for constructing a minimal cost path from a source node to a destination node considering a mobile network. Notice that this mathematical model yields same results as the first one, but is mathematically better described.

The variables for this new approach are described in the Table 2. The sets and parameters are the same from the Table 1.

Table 2: Variables description for the Second Approach

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 <i>Destination</i> (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 <i>Destination</i> (Binary variable).
$D_{jl}$	Determines if the node $j$ is selected at the destination state $l$ (Binary variable).
$DS_l$	Determines if the state $l$ is selected as a destination state (Binary variable).

Next, the second mathematical model is described.

$$\min \sum_{itjul} C_{it}^{jul} X_{it}^{jul} \quad (6)$$

Subject to:

$$\sum_{l>1} D_{jl} = 1 \quad \forall j \in N \quad (7)$$

$$\sum_l D_{jl} = 1 \quad \forall j \in N \mid j \neq Destination \quad (8)$$

$$D_{jl} * DS_l = D_{jl} \quad \forall j \in N; \quad \forall l \in S \quad (9)$$

$$\sum_l DS_l = 1 \quad (10)$$

$$DS_l = 0 \quad \forall l \in S \mid l = 1 \quad (11)$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jul} Y_{im} D_{jl} = DS_l \quad \forall l, m \in S \mid m = l - 1 \quad (12)$$

$$DS_l \sum_i Y_{im} = DS_l \quad \forall l, m \in S \mid m \leq l \quad (13)$$

$$DS_l \sum_i Y_{im} = 0 \quad \forall l, m \in S \mid m > l \quad (14)$$

$$\sum_{i \in N} \sum_{t \in S} \sum_{u \in S} \sum_{l \in S} X_{it}^{jul} = 1 \quad \forall j \in N \mid j = Destination \quad (15)$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jum} Y_{in} Y_{jm} = DS_l \quad \forall l, m, n \in S \mid m > 1 \wedge m = l \wedge n = m - 1 \quad (16)$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jum} = DS_l \quad \forall l, m \in S \mid m \leq l \quad (17)$$

$$\sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jul} Y_{jl} = 1 \quad \forall i \in N \mid i = Source \quad \forall l \in S \mid l = 1 \quad (18)$$

The equation 6 corresponds to the objective function, which will try to find the  $X_{it}^{jul}$  variables with the less possible cost  $C_{it}^{jul}$ . The previous expressions are explained in the following items:

- Destination State Constraints (from 7 to 15 ): The following expressions are referred to the Destination State, that is, the state at which the Destination node is found at the minimum possible cost.
  - Defining  $D_{jl}$ :  $D_{jl}$  allows to obtain the Destination State  $l$  at which the Destination node  $j$  is found at the minimum possible cost. The expression 7 avoids that  $D_{jl}$  will be one at the first state. The equation 8 avoids  $D_{jl}$  will be one for a node different from the destination node.
  - Defining  $DS_l$ :  $DS_l$  allows to extract only the Destination State  $l$  at which the Destination node is found at the minimum possible cost. The expression 9 allows to know the state  $l$  at which  $D_{jl}$  was selected. The equation 10 indicates that only one destination state is possible. In the expression 11 we assume it is not possible that the destination state will be the first state.

- Selecting the forwarding node: The forwarding node indicates the node selected at each state for constructing the minimum cost path. The expressions 12 and 13 restricts to one the number of  $Y_{jl}$  for each State less than the Destination State. The equation 14 restricts to zero the number of  $Y_{jl}$  for each State higher than the Destination State. The expression 15 indicates that it is possible only one link to the Destination node for all states, that is, only one state is selected, and for the rest of the states, the link must be zero.
- Intermediate State Constraints: These constraints allow selecting the predecessor node  $Y_{im}$  based on the current forwarding node  $Y_{jl}$ . In order to understand what these two types of nodes means, let's see an example. If we have a link between the nodes 1 and 2 in the direction from 1 to 2, the current forwarding node is 2 and the predecessor node is 1. The expression 16 allows to select the predecessor node at the intermediate states, where intermediate states refers to the states between the Destination and the Source States. The equation 17 restricts to one the number of  $X_{it}^{jul}$  for each state equal or less than the Destination State.
- Source State Constraint: The Source State indicates the State at which the Source node starts to construct the minimum cost path. The expression 16 restricts to one the number of  $X_{it}^{jul}$  for the Source state.
- Defining the First State Solution Constraint: All the constraints described above allow to find the minimum cost path between a Source node and a Destination node through several Network States. However, up to now our model does not consider the Destination State can be the first network state. For this reason it is necessary to apply the following post-processing pseudocode:

---

**Algorithm 2** Post-processing pseudocode
 

---

```

1: parameters Source, Destination
2: minSolution = MathModel(Source, Destination)
3: costFirstState =  $C_{it}^{jul}$  |  $i = \text{Source}, j = \text{Destination}, t = u = l = 1$ 
4: if costFirstState < minSolution then
5:   minSolution = costFirstState
6: end if

```

---

This pseudocode basically indicates that if the cost between the Source and the Destination node is less than the solution found by the mathematical model, then the solution is at the first state, otherwise the solution is given by the mathematical model.

## 5 Algorithms proposals

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

### 5.1 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.

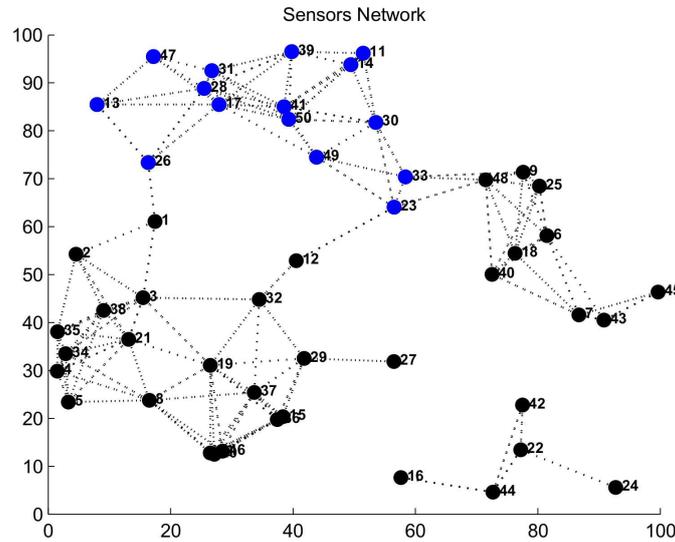


Figure 3: Problem scenario

In the Figure 3 is presented the following example. 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 pseudocodes of these two methods are described below.

---

**Algorithm 3** Destination node algorithm

---

```

1:  $d = destination$ 
2:  $r = number\ of\ rounds$ 
3:  $dn = neighbours(d)$  ▷  $dn$ : destination neighbours
4: while  $r > 0$  do
5:    $dn = neighbours(dn)$ 
6:    $sendControlMessage(dn)$ 
7:    $r = r - 1$ 
8: end while
    
```

---

The Algorithm 2 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.

The Algorithm 3 corresponds to the 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 has received the message from the Source node, it has to find a new neighbour node for passing the message. This process is performed for the next selected neighbour nodes until a connected node or the destination node has been reached. Remember

that a connected node will allow us to reach rapidly the Destination node in such case that a message arrives to them. This is possible since the connected node is linked directly to the Destination node or to other connected node which possibly will be connected directly to the Destination node. Once a neighbour node is selected for sending the message, this node is added to the array *path*. Then, when the Destination is reached, the array *path* contains the message's traceability from the Source node to the Destination node.

---

**Algorithm 4** Forwarding node algorithm for SIPCOS
 

---

```

1: parameter s                                     ▷ Source Node
2: parameter d                                     ▷ Destination Node
3: parameter dN                                    ▷ Destination Neighbours Nodes
4: array path = []                                ▷ The Building Communication Path
5: variable fn = s                                ▷ fn: Forwarding Node
6: function FN(i)                                  ▷ FN: Forwarding Neighbours Function
7: minCost = ∞
8: DestinationNeighbourFound = 0
9: array path = [path fn]
10: while message ∉ dN ∨ d do
11:   forwardingNeighbours = neighbours(fn)
12:   for i = 1 to forwardingNeighbours do
13:     if FN(i) ∈ dN then
14:       DestinationNeighbourFound = 1
15:       sendMessage(FN(i))
16:       path = [path FN(i)]
17:     end if
18:   end for
19:   if DestinationNeighbourFound == 0 then
20:     for i = 1 to forwardingNeighbours do
21:       neighbourCost = cost(fn, FN(i))
22:       if neighbourCost < minCost then
23:         fn = FN(i)
24:         sendMessage(fn)
25:         path = [path fn]
26:       end if
27:     end for
28:   end if
29: end while
30: if message ∈ dN then
31:   dN = neighbours(dN)
32:   if Destination ∈ dN then
33:     sendMessage(d)
34:     path = [path d]
35:   else
36:     sendMessage(dN)
37:     path = [path dN]
38:   end if
39: end if

```

---

## 5.2 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 5** Forwarding node algorithm for SIP

```

1: parameter  $s$ 
2: parameter  $d$ 
3: array  $path = []$ 
4: variable  $fn = s$ 
5: function  $FN(i)$ 
6:  $minCost = \infty$ 
7: array  $path = [path \ fn]$ 
8:  $DestinationFound = 0$ 
9: while  $message \notin d$  do
10:    $forwardingNeighbours = neighbours(fn)$ 
11:   for  $i = 1$  to  $forwardingNeighbours$  do
12:     if  $forwardingNeighbours(i) == d$  then
13:        $DestinationFound = 1$ 
14:        $sendMessage(d)$ 
15:        $path = [path \ FN(i)]$ 
16:     end if
17:   end for
18:   if  $DestinationFound == 0$  then
19:     for  $i = 1$  to  $forwardingNeighbours$  do
20:        $neighbourCost = cost(forwardingNode, FN(i))$ 
21:       if  $neighbourCost < minCost$  then
22:          $forwardingNode = FN(i)$ 
23:          $sendMessage(fn)$ 
24:          $path = [path \ fn]$ 
25:       end if
26:     end for
27:   end if
28: end while
29: if  $message \in d$  then
30:    $path = [path \ d]$ 
31: end if

```

## 6 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 following tables:

Table 3: 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

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 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

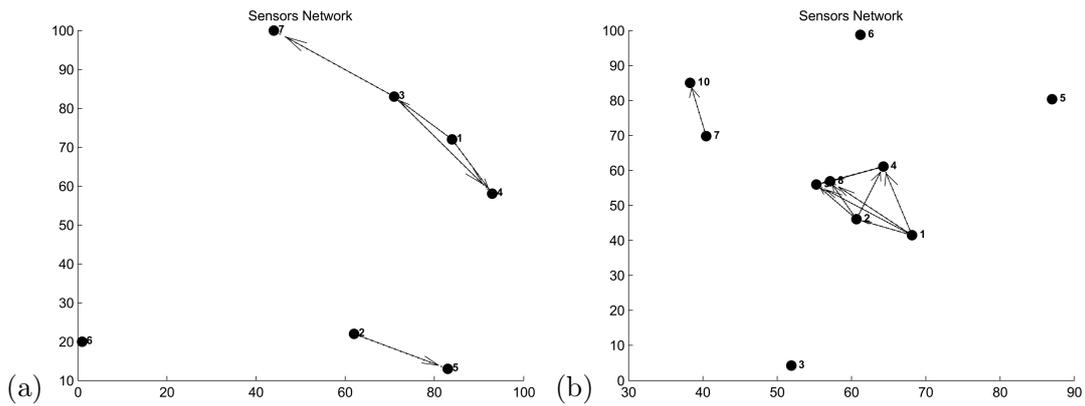


Figure 4: Problem definition: (a) Scenario 1; (b) Scenario 2

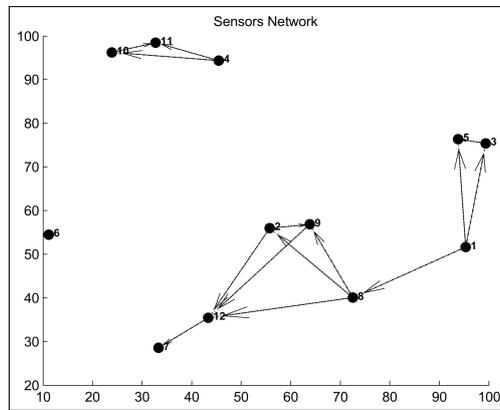


Figure 5: Scenario 3

Table 4: 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 5: 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

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.

## 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.

## Bibliography

- [1] I. F. Akyildiz and M. C. Vuran (2010); *Wireless Sensor Networks*, Wiley, 2010.
- [2] J. Zheng and A. Jamalipour (2009); *Wireless Sensor Networks: A Networking Perspective*, Wiley-IEEE Press, 2009.
- [3] A. A. Ahmed (2013); An enhanced real-time routing protocol with load distribution for mobile wireless sensor networks, *Computer Networks*, 57(6):1459-1473.
- [4] A.A. Ahmed (2007); *Real-Time Wireless Sensor Networks*, University of Virginia, 2007.
- [5] B. Buchli, F. Sutton, J. Beutel (2012); GPS-Equipped Wireless Sensor Network Node for High-Accuracy Positioning Applications, *Wireless Sensor Networks Lecture Notes in Computer Science*, Springer, 7158:179-195.
- [6] S. Li, X. Ma, X. Wang, M. Tan (2011); Energy-efficient multipath routing in wireless sensor network considering wireless interference, *Journal of Control Theory and Applications*, 9(1):127-132.
- [7] G. M. de Araujo, J. Kaiser and L. B.Becker (2014); Genetic Machine Learning Approach for Link Quality Prediction in Mobile Wireless Sensor Networks, *Cooperative Robots and Sensor Networks*, 1-14.
- [8] G. M. de Araujo, J. Kaiser, L. B.Becker (2012); An Optimized Markov Model to Predict Link Quality in Mobile Wireless Sensor Networks, *Procedia Computer Science*, 10:1100-1105.
- [9] J. A. Torkestani Young (2012); Mobility prediction in mobile wireless networks, *Journal of Network and Computer Applications*, 35(5):1633-1645.
- [10] G. A. Montoya and Y. Donoso (2016); A Delay-Sensitive Mathematical Model Approach and a Distributed Algorithm for Mobile Wireless Sensor Networks, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore doi: 10.1109/ICCCC.2016.7496736, 45-50.