

A Prediction Algorithm based on Markov Chains for finding the Minimum Cost Path in a Mobile WSNs

G.A. Montoya, Y. Donoso

Germán A. Montoya*, **Yezid Donoso**

Systems and Computing Engineering Department
Universidad de Los Andes, Bogotá, Colombia

*Corresponding author: ga.montoya44@uniandes.edu.co
ydonoso@uniandes.edu.co

Abstract: In this paper we propose the usage of a prediction technique based on Markov Chains to predict nodes positions with the aim of obtain short paths at minimum energy consumption. Specifically, the valuable information from the mobility prediction method is provided to our distributed routing algorithm in order to take the best network decisions considering future states of network resources. In this sense, in each network node, the mobility method employed is based on a Markov model to forecast future RSSI states of neighboring nodes for determining if they will be farther or closer within the next steps. The approach is evaluated considering different algorithms such as: Distance algorithm, Distance Away algorithm and Random algorithm. In addition, with the aim of performing comparisons against optimal values, we present a mathematical optimization model for finding the minimum cost path between a source and a destination node considering all network nodes are mobile. This paper is an extended variant of [8]^a.

Keywords: Markov model, RSSI, MWSN.

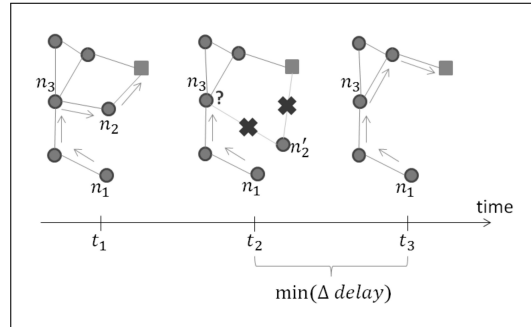
^aPartially reprinted and extended, with permission based on Licence Number 4484761270350 © IEEE, from "2018 7th International Conference on Computers Communications and Control (ICCCC)."

1 Introduction

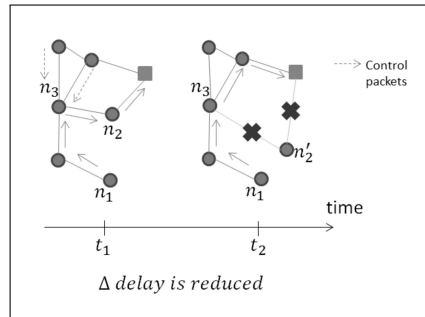
The advances of WSN have allowed attaching the sensors to an entity such as objects, animals or humans, to monitor a physical variable in its environment. However, the sensors are equipped with limited batteries whereby it is required to implement energy efficient routing techniques to extend the lifetime of the sensors as much as possible [3, 10]. In addition, communication disruptions caused by mobility in wireless sensor networks introduce undesired delays that affect the network performance in delay sensitive applications, such as health monitoring applications. Due to these kind of applications must provide a timeliness performance since they are dealing with health issues such as illness and continuous medical supervision, these applications must operate with minimum possible delay. In other words, a base station should no experiment delays from the information collected by the sensors [1, 2].

Given the scenario described above, a possible solution would consist to implement energy efficient routing techniques considering the sensor position to know the mobility level of the network. Based on this information, it is possible to determine the nodes that considerably affect the communication performance of the network. Some of these solutions propose the usage of sensors equipped with GPS devices, called GPS non-free approaches. However, these GPS non-free solutions have in most of cases drawbacks such as high implementation costs, delays for acquiring position information and non-accurate position information [4]. In addition, these types of solutions require an extra chip for the GPS [7], whereby more energy consumption is experimented. For these reasons, our work will not take into account sensors equipped with

GPS devices. Thus, in order to be aware of the network mobility we are going to use RSSI measurements, which indicates an approximated distance between two nodes.



(a) Problem



(b) Solution

Figure 1: Problem definition

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 carrying the 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. The communication reestablishment between n_1 and the base station can be perfectly performed using routing techniques, but at the expense of introduce an undesired delay in this communication path. 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 health monitoring, this disadvantage might mean a very low network performance.

Given the problem above, our proposal consists to use a predicting technique which is described in the figure 1.b) [5,6]. It consists of the same situation showed in the Figure 1.a), but in this case, at time t_1 , the node n_3 receive information that indicates the node n_2 will rapidly be away from its communication range, at time t_2 . Given this information, n_3 , at time t_1 , is also analyzing a possible candidate which could replace n_2 , in the case n_2 fails in a future time. If, indeed, at time t_2 the node n_2 fails because it has moved away from n_3 , this node at time t_2 can promptly reestablish the communication path between n_1 and the base station, reducing the delay described in the Figure 1.a).

In order to solve the problem presented above, we propose to use a predicting method based on a Markov Model for estimating future RSSI states for a node with the aim of minimizing the delay

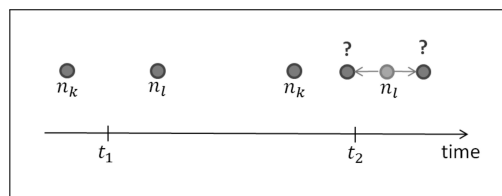
experimented in the network. In this sense, our approach will be evaluated considering a Gauss-Markov mobility mode [9] where the mobility nodes can be considered predictable in order to test our prediction algorithm. Our work pretends to show an increasing network performance in terms of end-to-end delay and energy consumption against different algorithms such as: Distance algorithm, Distance Away algorithm and Random algorithm, which will be described in detail in the next sections. This paper is an extension of the work presented in ICCCC2018 [8]. In this sense, the extension consists to present a mathematical optimization model for finding the minimum cost path between a source and a destination node considering a mobile network with the aim of performing comparisons between optimal values and the algorithms proposed.

The remainder of the paper is organized as follows: Section 2 presents the prediction algorithm based on Markov Chains, that is, how the states are defined and the probability from going to one state to another. Section 3 shows the mathematical optimization model proposed, the objective function and the constraints. Finally, section 5 and 6 show the results and conclusions respectively.

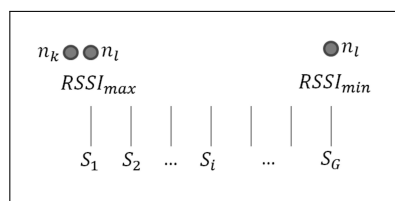
2 Prediction algorithm formulation

In order to solve the problem presented above, we propose to use a predicting method based on a Markov Model for estimating future RSSI states of a node with the aim of minimizing the delay experimented in the network. For this purpose, a detailed explanation, supported with the following figures, will be presented.

In relation to the Figure 2.a), suppose we have a network compound of two nodes: n_k and n_l , where n_l is a neighboring node of n_k . There are two times, t_1 and t_2 , at which our little network is evolving in time. At time t_1 the node n_l is located at certain distance from n_k . However, at time t_2 we want to predict if n_l will be farther or closer (or at the same distance in t_1) from n_k .



(a) Possible movement of n_l



(b) RSSI States

Figure 2: Defining Markov states - 1

Respect to the Figure 2.b), there is a minimum and maximum distance at which n_l can be located in order to establish a communication link with n_k . At the minimum distance, n_l will have a maximum RSSI, $RSSI_{max}$, and, at the maximum distance, n_l will have a minimum RSSI, $RSSI_{min}$. At t_2 , n_l could be located at any distance between $RSSI_{min}$ and $RSSI_{max}$.

Our goal consists to estimate the location between $RSSI_{min}$ and $RSSI_{max}$ at which n_l will be in a future time (in this case, t_2). Theoretically, there are infinite locations between $RSSI_{min}$ and $RSSI_{max}$, but for our model we assume discrete locations equitably spaced. These possible locations, at which n_l could be, are called states. In this sense, at a future time t_2 , n_l could be at S_1 , S_2 , S_r or S_G , where G is the maximum number of states. The initial probability of n_l for being at any state S_i is $1/G$, which is called *Initial Probability Distribution of set S* (π), can be expressed as follows:

$$\pi = \{P_{s_1}, P_{s_2}, \dots, P_{s_G}\} \quad (1)$$

According to the Figure 2.a), suppose we want to know the probability to go from the the state S_2 to the state S_4 , which is calculated with the following expression:

$$P_{24} = \frac{N(S_2, S_4)}{\sum_{j=1}^G N(S_2, S_j)} \quad (2)$$

Where $N(S_i, S_j)$ is the number of times that the state S_i follows state S_j .

This expression can be extensible for the rest of probabilities, as it is indicated in the following expression:

$$P_{ij} = \frac{N(S_i, S_j)}{\sum_{j=1}^G N(S_i, S_j)} \quad (3)$$

In this sense, we have the probability to go from any state S_i to any state S_j . These probabilities can be expressed in a matrix, which is called *Transition Matrix*:

$$T = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1G} \\ P_{21} & P_{22} & \dots & P_{2G} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ P_{G1} & P_{G2} & \dots & P_{GG} \end{bmatrix} \quad (4)$$

In relation to the figure 3.b), suppose that in a current time t_1 , n_l is at state S_3 and we want to estimate the future state of n_l at a future time t_p . For this purpose, we can apply the following expressions:

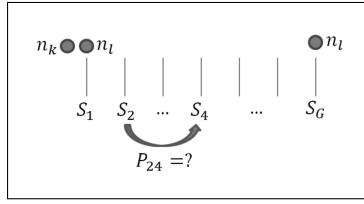
$$\pi_p = \pi * T^p \quad (5)$$

$$S_p = \max\{\pi_p\} \quad (6)$$

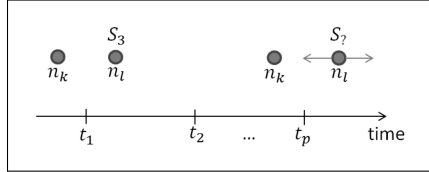
$$S_p = \max\{P_{s_1}, P_{s_2}, \dots, P_{s_G}\} \quad (7)$$

According to the expression 7, n_k can finally obtain the most probable future state at which n_l will be at a time t_p , and use this information for routing decisions in order to reduce the delay caused for probable communication disruptions in the future.

The present approach will be evaluated considering a Gauss-Markov mobility mode [9] where the mobility nodes can be considered predictable in order to test our prediction algorithm. Our work pretend to show an increasing network performance in terms of end-to-end delay and energy consumption against an approach without using a mobility prediction method and other approaches existent. Additionally, we will compare our algorithm results against a mathematical model optimization which minimizes energy consumption considering delay and network resources constraints.



(a) Probability to go from state S_2 to S_4 .



(b) Predicting the future state of n_l .

Figure 3: Defining Markov states - 2

3 Mathematical model formulation

In this section our problem is enunciated and described in detail from a mathematical optimization point of view, as well as some assumptions are shown in order to simplify our proposed mathematical model. Remember that, in addition to the prediction algorithm approach, we present a mathematical optimization model for finding optimal values with purpose of performing valuable comparisons against the others algorithms described in this work. Now, we present the details and assumptions of our mathematical model proposal.

Based on the Figure 4, 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 4.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 4. For example, n_{11} can communicate with n_{21} and n_{31} .

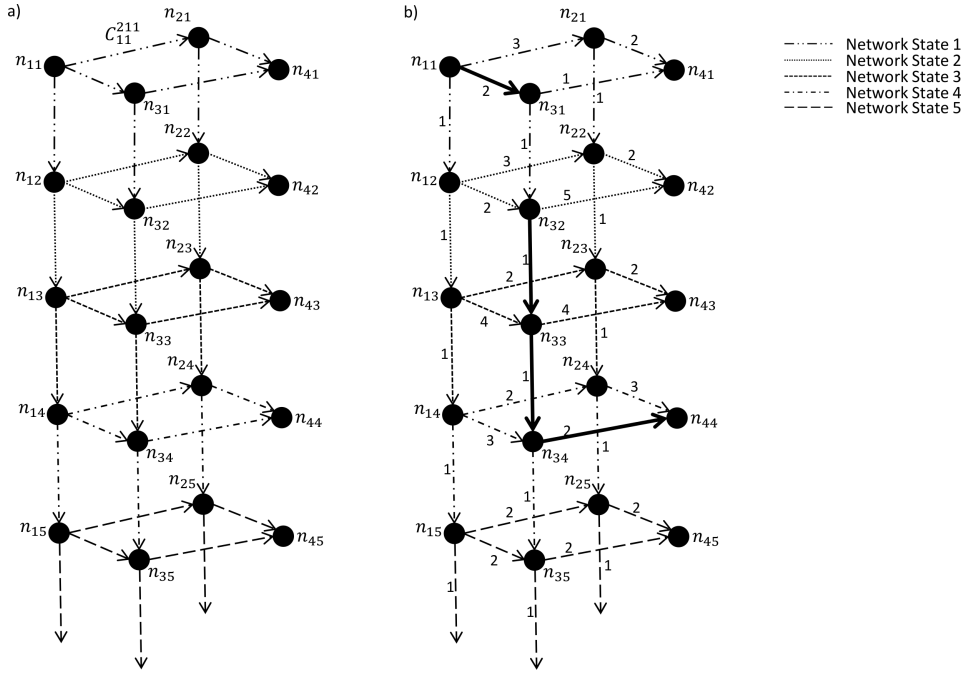


Figure 4: Problem scenario

- **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}^{211} , and denoted as C_{it}^{jul} , that is, this is the cost to carry a message from 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 4.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.

Following is presented our mathematical model proposed for constructing a minimal cost path from a source node to a destination node considering a mobile network. The sets, parameters and variables required for the mathematical model are described in the Table 1.

Table 1: Sets, parameters and variables description for the mathematical model

Sets	Description
N	Set of network nodes.
S	Set of network states.
Parameters	Description
C_{it}^{juld}	Link cost at the state l from the node i at the state t to the node j at the state u .
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, our mathematical model is described.

$$\min \sum_{itjul} C_{it}^{jul} X_{it}^{jul} \quad (8)$$

Subject to:

$$\sum_{l>1} D_{jl} = 1 \quad \forall j \in N \quad (9)$$

$$\sum_l D_{jl} = 1 \quad \forall j \in N \mid j \neq \text{Destination} \quad (10)$$

$$D_{jl} * DS_l = D_{jl} \quad \forall j \in N; \quad \forall l \in S \quad (11)$$

$$\sum_l DS_l = 1 \quad (12)$$

$$DS_l = 0 \quad \forall l \in S \mid l = 1 \quad (13)$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jul} Y_{im} D_{jl} = DS_l \quad \forall l, m \in S \mid m = l - 1 \quad (14)$$

$$DS_l \sum_i Y_{im} = DS_l \quad \forall l, m \in S \mid m \leq l \quad (15)$$

$$DS_l \sum_i Y_{im} = 0 \quad \forall l, m \in S \mid m > l \quad (16)$$

$$\sum_{i \in N} \sum_{t \in S} \sum_{u \in S} \sum_{l \in S} X_{it}^{jul} = 1 \quad \forall j \in N \mid j = Destination \quad (17)$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jum} Y_{in} Y_{jm} = DS_l \quad \forall l, m, n \in S \mid m > 1 \wedge m = l \wedge n = m - 1 \quad (18)$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jum} = DS_l \quad \forall l, m \in S \mid m \leq l \quad (19)$$

$$\sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jul} Y_{jl} = 1 \quad \forall i \in N \mid i = Source \quad \forall l \in S \mid l = 1 \quad (20)$$

The equation 8 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 9 to 17): 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 9 avoids that D_{jl} will be one at the first state. The equation 10 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 11 allows to know the state l at which D_{jl} was selected. The equation 12 indicates that only one destination state is possible. In the expression 13 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 14 and 15 restricts to one the number of Y_{jl} for each State less than the Destination State. The equation 16 restricts to zero the number of Y_{jl} for each State higher than the Destination State. The expression 17 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 18 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 19 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 18 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 1 Post-processing pseudocode

```

1: parameters Source, Destination
2: minSolution = MathModel(Source, Destination)
3: costFirstState = Citjul | i = Source, j = 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.

4 Implementation

We have designed a Mobile Wireless Sensor Network Simulator in MATLAB, which has the following basic network components:

- *Destination node:* it is the final node that will receive a data message. In our simulations this node will always be the last network node.
- *Source node:* This node will have a data message, which must arrive to the destination node. In our simulations this node will always be the first network node.
- *Connected node:* If a message arrive to this node, this node knows the path to achieve the destination node.

In order to test the Prediction technique above, our simulator is compound of the following main processes:

- *Forwarding node selection:* When a node has a data message, this process consists to select properly a neighbour node as a forwarding node, which is selected according to the following priorities:
 - If among the neighbour nodes there is the destination node, then, the forwarding node is the destination node.
 - If among the neighbour nodes there is not the destination node, but there is a connected node, then, the forwarding node is the connected node.
 - If among the neighbour nodes there is not a destination node neither a connected node, then, the forwarding node is a node given by the Prediction method.

- *Sink refreshing*: This process consists to determine which nodes will be connected nodes at each certain period. This refreshing process is required due to network mobility, since it causes that connected nodes established in a previous state period, they will not possibly be connected nodes in the next period.
- *Loop detection*: It is important that a message can achieve the destination node, whereby it is necessary avoiding the message fall into a loop.
- *Prediction at each k -state*: At each network state the *Transition Matrix* (T) is calculated for all network nodes, except the destination node. Remember that this Transition Matrix stores the probability of each node to be at certain distance level respect to their neighbour nodes.
- *Prediction for selecting a forwarding node*: As we say before, if among the neighbour nodes there is not a destination node neither a connected node, then, the forwarding node is a node given by the Prediction method. This forwarding node is selected based on the information given by the Transition Matrix.

In order to test the Prediction algorithm performance, we have designed more algorithms with the aim of doing comparisons and obtain valuable information. Next, there is a description of each algorithm respect to its forwarding node process selection:

- *Distance Algorithm*: Considering there is not a destination or a connected node among the neighbour nodes of a current node, the forwarding node is the node with the shortest distance to the current node. The current node is the one that currently has a message that must arrive to the destination node.
- *Distance Away Algorithm*: Considering there is not a destination or a connected node among the neighbour nodes, the forwarding node is the node with the longest distance to the current node.
- *Prediction Algorithm*: Considering there is not a destination or a connected node among the neighbour nodes, the forwarding node is the node with the best probability to be near to the current node.
- *Prediction Away Algorithm*: Considering there is not a destination or a connected node among the neighbour nodes, the forwarding node is the node with the best probability to be far away to the current node.
- *Random Algorithm*: Considering there is not a destination or a connected node among the neighbour nodes, the forwarding node is a random node.

5 Results

In this section we will present the main results for the different algorithms showed in the previous sections. The metrics used for showing these results are: *Energy Consumption* and *Hops*. The *Energy Consumption* metric indicates the energy wasted by all the network nodes until the destination node is found. The *Hops* metric indicates the amount of hops needed to find the destination node. These two metrics are showed versus the number of network nodes. In addition, in order to obtain valuable statistical results, the performance evaluation of each algorithm and the mathematical model was made for 10000 tests for each network size. The next

figures show the performance of the different algorithms and the mathematical model proposed for finding the minimum cost path in the network.

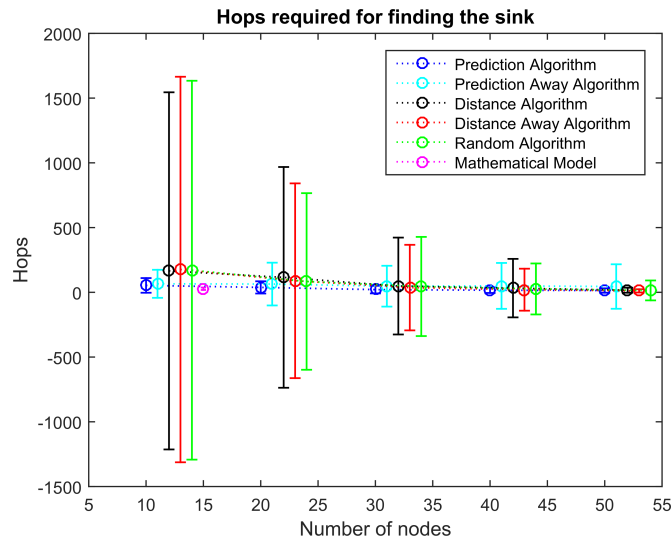


Figure 5: Hops performance along the network size

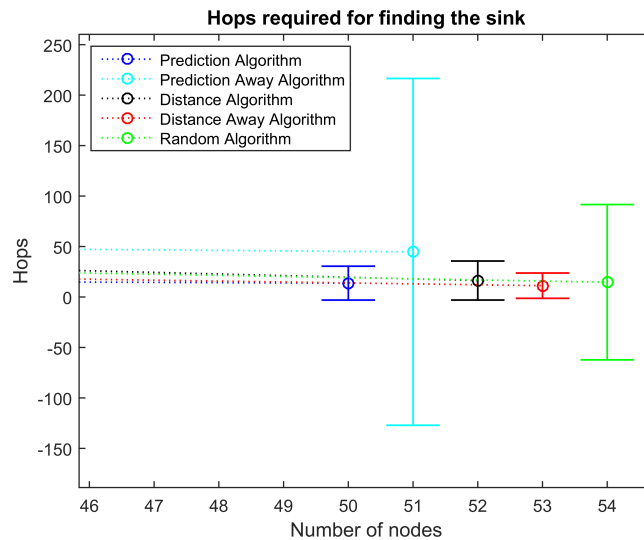


Figure 6: Hops for 50 nodes

The figure 5 shows the hops performance by the different algorithms along the network size. From this figure, as the size decreases the performance of Distance, Distance Away and Random algorithms gets worse because they require more hops to find the destination node. By contrast, the performance of Prediction and Prediction Away algorithms is better than the other ones because it requires less hops to find the destination node. This can be explained by the usage of prediction techniques, which offer more reliable paths. The following figures are focused in each network size.

The figure 6 shows the algorithms hops performance for 50 nodes. The number of hops for each algorithm is presented in the table 2. The best performance is obtained by the Distance algorithm, while the Prediction algorithm is second best because the network size is big (50 nodes), allowing more path alternatives for the Distance Away algorithm to find faster the destination node.

Table 2: Hops for 50 nodes

	Hops	Ranking
Prediction Algorithm	13.72	2
Prediction Away Algorithm	44.71	5
Distance Algorithm	16.30	4
Distance Away Algorithm	11.20	1
Random Algorithm	14.66	3

The figure 7 shows the algorithms hops performance for 40 nodes. The number of hops for each algorithm is presented in the table 3. The best performance is obtained by the Prediction algorithm because the number of nodes begins to decrease compared with the 50 nodes scenario, generating less path alternatives for the others algorithms and, then, thanks to the reliable feature given by the prediction technique, this algorithm can achieve faster the destination node.

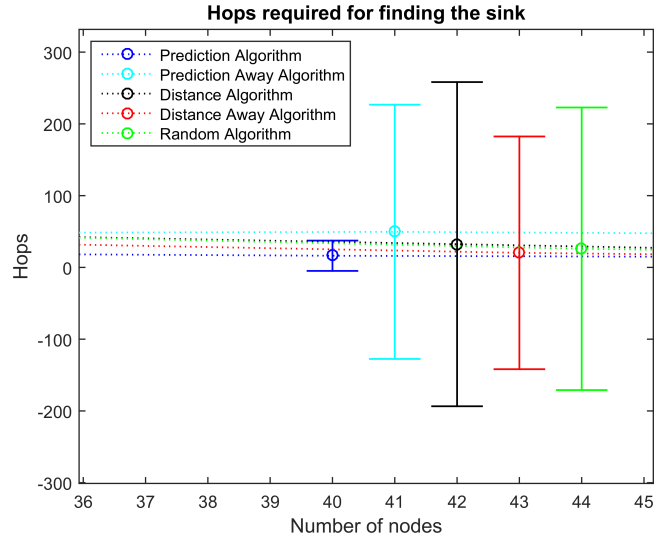


Figure 7: Hops for 40 nodes

Table 3: Hops for 40 nodes.

	Hops	Ranking
Prediction Algorithm	16.24	1
Prediction Away Algorithm	49.61	5
Distance Algorithm	32.38	4
Distance Away Algorithm	20.31	2
Random Algorithm	25.95	3

The figure 8 shows the algorithms hops performance for 30 nodes. The number of hops for each algorithm is presented in the table 4. The best performance is obtained again by the Prediction algorithm for the same reason as the previous figure. The less size of the network, the less path alternatives will have the rest of algorithms.

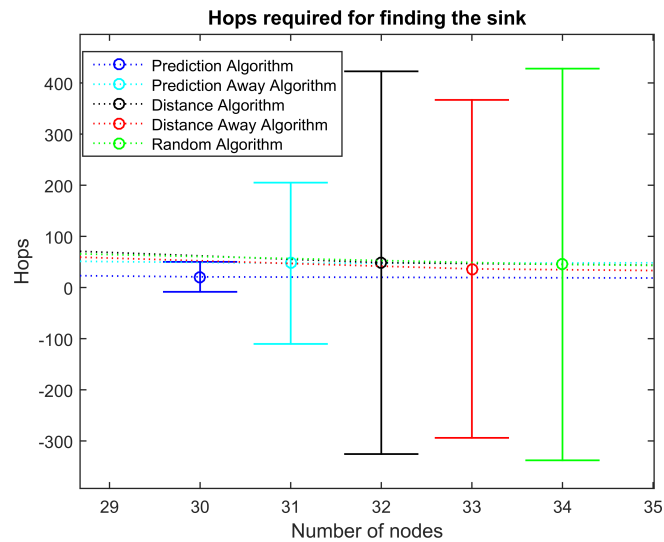


Figure 8: Hops for 30 nodes

Table 4: Hops for 30 nodes

	Hops	Ranking
Prediction Algorithm	20.86	1
Prediction Away Algorithm	47.38	4
Distance Algorithm	48.53	5
Distance Away Algorithm	36.42	2
Random Algorithm	45.05	3

The figure 9 shows the algorithms hops performance for 20 nodes. The number of hops for each algorithm is presented in the table 5.

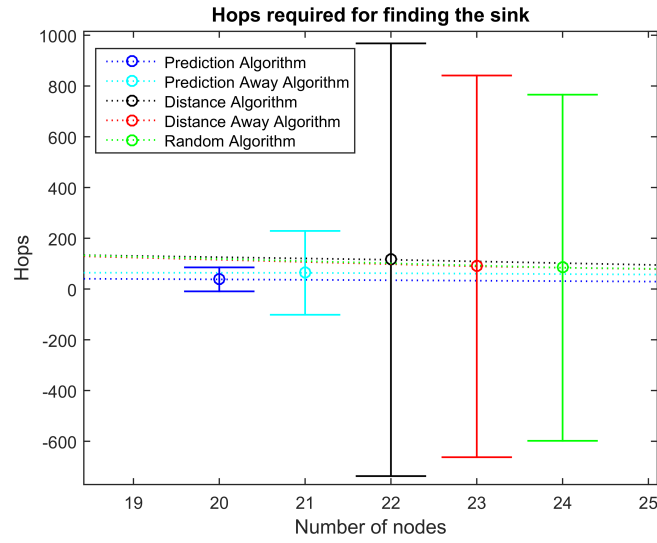


Figure 9: Hops for 20 nodes

Table 5: Hops for 20 nodes

	Hops	Ranking
Prediction Algorithm	37.99	1
Prediction Away Algorithm	63.75	2
Distance Algorithm	115.38	5
Distance Away Algorithm	89.30	4
Random Algorithm	83.85	3

The figure 10 shows the algorithms hops performance for 10 nodes. The number of hops for each algorithm is presented in the table 6. Here we can notice the large difference in terms of hops of using prediction techniques compared with not-using prediction techniques. This means that if our network has few nodes and, as a consequence, it is more difficult to find a path to the destination node, our prediction algorithm is capable of obtain a large advantage respect the others algorithms for finding the destination node. This advantage is represented in the hop different respect to the second algorithm in the ranking, which is 25.75 hops of difference. This comparison is among the Prediction and Prediction Away algorithms. However, if the comparison is done between the Prediction algorithm and the best algorithm that does not use prediction techniques (the Random Algorithm), the advantage of using the Prediction algorithm is even higher (45.85 hops). This indicates using prediction techniques are suitable when finding paths is a critical task, that is, when the network is compound of few nodes. Notice that in addition there are presented the results for the mathematical model, which obviously presents the best performance, showing a hops performance difference of 25.81 respect to the Prediction algorithm. Notice that the mathematical model tests where suitable in terms of time execution and memory usage for a maximum of 15 nodes. For instance, a test of 30 nodes or even 20 nodes, the mathematical model solution unfortunately never ended. For this reason, only solutions for 10 nodes is provided to be compared with the prediction routing algorithm and no-prediction routing algorithms.

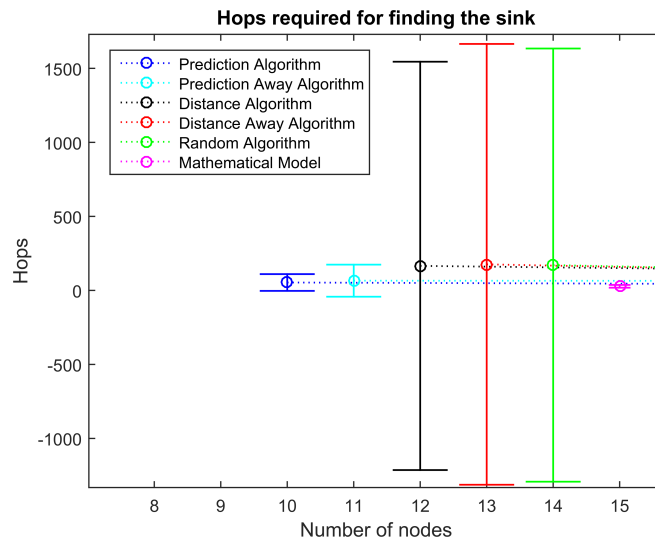


Figure 10: Hops for 10 nodes

Table 6: Hops for 10 nodes

	Hops	Ranking
Prediction Algorithm	53.38	1
Prediction Away Algorithm	65.75	2
Distance Algorithm	165.62	3
Distance Away Algorithm	176.07	5
Random Algorithm	170.92	4
Mathematical Model	27.57	

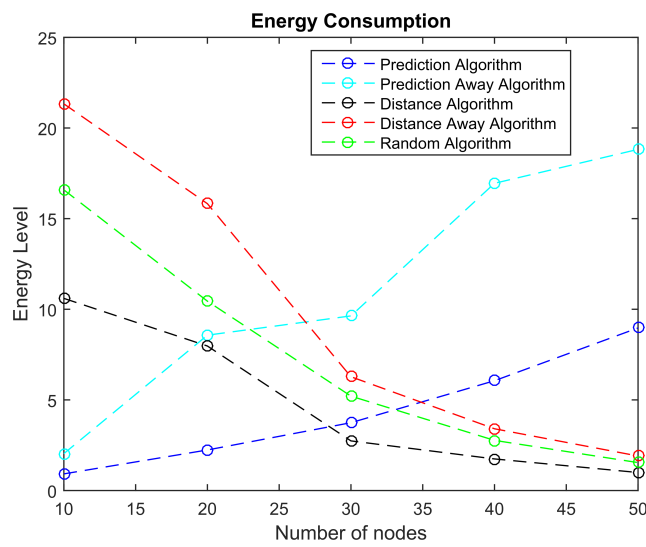


Figure 11: Energy consumption of the network

The figure 11 shows the algorithms energy consumption performance for different network size. This figure indicates that the Prediction Algorithm is suitable for low size network, showing that it starts to be efficient in terms of energy consumption from almost 30 nodes to 10 nodes.

6 Conclusions

We proposed the usage of a prediction technique in the context of a mobile wireless sensor network with the aim of the shortest path possible from a source node to a destination node. Employing this technique allowed to building the most reliable path for finding the destination node and at the same time it allowed to obtain the shortest path to the destination node. In other words, the reliability offered by the prediction technique allowed to select the most stable forwarding nodes in terms of their network connectivity. In this sense, through the prediction technique it was less likely that a data message would be in isolated network zones, and then, there was a higher probability for reaching the destination node by the data message. For this reason, when the number of network nodes was scarce, 10 or 20 nodes, the prediction algorithm performance was too high in comparison with the rest of the algorithms, obtaining 45.85 and 112.24 hops of difference with the second best no-prediction algorithm for the 10 and 20 nodes of network size. The impact of this finding is very interesting. Suppose a cattle application where the network nodes (20 nodes) changes each 100 miliseconds. This means that if we use the prediction algorithm, a data message will reach the destination node 11.22 seconds faster than the second best no-prediction algorithm. This time, 11.22 seconds, could be a significant advantage in delay sensitive applications where the timeliness is an imperative factor.

In terms of energy consumption, a prediction technique is suitable for scarce networks (10, 20 or 30 nodes) because the energy consumption was the less than the rest of algorithms. This energy performance besides to the hops performance make the prediction algorithm totally suitable for scarce networks, that is, MWSN applications where the number of nodes is not too high and it is required data messages arrive to the destination nodes as soon as possible.

Bibliography

- [1] Ahmed A. A. (2013); An enhanced real-time routing protocol with load distribution for mobile wireless sensor networks, *Computer Networks*, 57, 2013.
- [2] Ahmed A. A. (2007); *Real-Time Wireless Sensor Networks*, University of Virginia, 2007.
- [3] Akyildiz, I. F.; Vuran, M. C. (2010); *Wireless Sensor Networks*, Vol. 4, John Wiley & Sons, Hoboken, 2010.
- [4] Buchli, B.; Sutton, F.; Beutel, J. (2012); GPS-Equipped Wireless Sensor Network Node for High-Accuracy Positioning Applications, *Wireless Sensor Networks Lecture Notes in Computer Science*, Springer, 7158, 179-195, 2012.
- [5] De Araujo, G. M.; J. Kaiser, J.; Becker, L. B. (2012); An Optimized Markov Model to Predict Link Quality in Mobile Wireless Sensor Networks, *Computers and Communications*, 307-312, 2012.
- [6] De Araujo, G. M.; J. Kaiser, J.; Becker, L. B. (2014), Genetic Machine Learning Approach for Link Quality Prediction in Mobile Wireless Sensor Networks, *Cooperative Robots and Sensor Networks*, 1-14, 2014.

- [7] Li, S.; Ma, X.; Wang, X.; Tan, M. (2011); Energy-efficient multipath routing in wireless sensor network considering wireless interference, *Journal of Control Theory and Applications*, 9(1), 127-132, 2011.
- [8] Montoya, G. A.; Donoso, Y. (2018); A Prediction Algorithm based on Markov Chains for finding the Minimum Cost Path in a Mobile Wireless Sensor Network, *Proceedings of the 7th International Conference on Computers Communications and Control*, IEEE, 169 - 175, 2018.
- [9] Torkestani, J. A. (2012); Mobility prediction in mobile wireless networks. *Journal of Network and Computer Applications*, 35(5), 1633-1645, 2012.
- [10] Zheng, J.; Jamalipour, A. (2009); *Wireless Sensor Networks: A Networking Perspective*, Wiley, 2009.
- [11] [Online] A Community Resource for Archiving Wireless Data At Dartmouth.
<http://crawdad.org/>.