

Resume

Jean-Francois COUCHOT, Khaled DAHER

April 30, 2015

1 Networks generation

1.1 Description

In this section, and at first, we created a tool that generates WWSN networks of n nodes each, randomly located in a square area of 50m x 50m. The node number n is the Sink, the others are video sensors and all nodes can send to the Sink. Each node has a transmission range of 30 meters, so in our tool we are seeking to maintain these conditions so that the network is connected and the nodes that can communicate are distant less than 30 meters.

1.2 Networks generator algorithm

```
Fill the link matrix  $a_{il}$  by zeros
Graph_connected = false
while (Graph_connected = false) do
    Generation of random coordinates of each node between 0 and 50 meters
    Calculate the distance between the nodes
    Test the distance between the nodes
    if (distance > 30 metres) then
        There is no link between these nodes
    end if
    Test if the graph is connected(algorithm of Dijkstra)
    if connected then
        Graph_connected = true
    end if
end while
Explore all feasible paths for each node to the sink
```

2 Implementation of networks

In this section, we test the network, generated in a network implementation tool (OMNET++) to visualize and evaluate the convergence of the distributed algorithm designed to maximize the lifetime of implemented WWSN network.

Networks are tested in synchronous mode, and next asynchronously. We collect the primal variables that characterize a WWSN network after 400 iterations, and finally an analysis is made of these results:

2.1 Synchronous mode

Synchronous mode consists of ensuring that each node makes as much iterations as the others in the same given time. This is done when a given node increments the number of iterations when receiving acknowledgment messages from all its neighbors.

2.2 Asynchronous mode

In asynchronous mode each node can make a number of iterations that differs from another node after a given time. Node increments the number of iterations when receiving acknowledgment messages from its neighbors, but not necessarily all of its neighbors, for example 50 % of them.

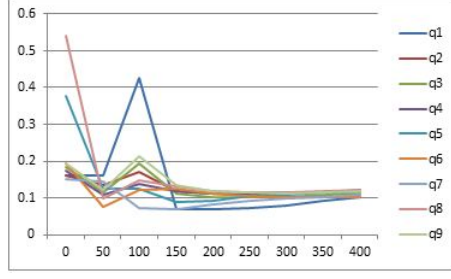
3 Results and Analyses

3.1 Interpretation

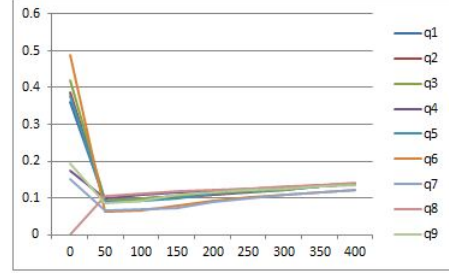
After testing a series of WWSN networks and based on the results obtained in our experiments, we noted that in some cases and in two modes: synchronous and asynchronous the values of the q_i variables, $1 \leq i \leq n$ in the nodes are almost the same as shown in fig1.a and fig1.b and in other cases these values are distant as in the fig2.a and fig2.b.

Furthermore, we noted in Fig 1.a (Synchronous mode) that the variation after 400 iterations between the q_i variables is 17.8 % and with a 0.00169497 convergence threshold for the dual function. Similarly in Fig 1.b (asynchronous mode) and for the same network the variation of q_i is 15.75 % with a convergence threshold of 7.17109e-005 for the dual function.

On the other hand when the q_i values are not near each other in some tests, and after 400 iterations (fig2.a, fig2.b) we noted that when applying the two synchronous and asynchronous modes, we almost got the same results, fig2.a: 38 % variation between the values with 0.000430264 as convergence threshold for the dual function and fig2.b: 38.8 % with 0.000229301.

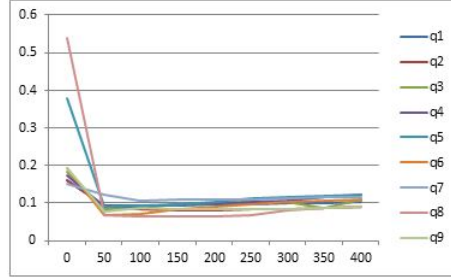


(a) synchronous mode

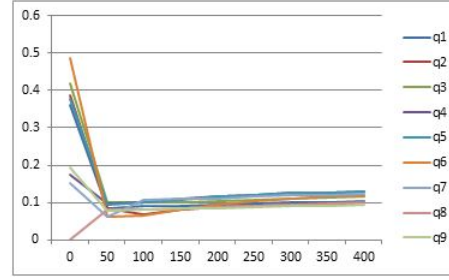


(b) asynchronous mode

Figure 1: q_i variations with the iterations(test1)



(a) synchronous mode



(b) asynchronous mode

Figure 2: q_i variations with the iterations(test2)

3.2 Conclusion

We can conclude from the above that the final results of the convergence conditions obtained are almost the same in the two synchronous and asynchronous modes. However, the synchronous mode is more efficient in terms of time to reach convergence.

In addition, Fig 1.b (asynchronous mode) shows the stabilization of q_i after 50 iterations which is not the case in the asynchronous mode (fig1.a) which there has been a remarkable fluctuation in the same interval.

So finally to achieve our main objective to maximize the lifetime of a network WWSN it is better now to consider the asynchronous mode to avoid time lost in the synchronous mode because each node waits for all the neighbors to respond for incrementing iterations and then extend the life of the network in the best delay.