Optimization of Connectivity in WSN using Fuzzy Logic

Lu Peng, Shannon Tao and Vivian Yang

Table of Contents

Introduction

Wireless sensor network (WSN)[1][2] is a promising technology nowadays. It is used in numerous applications, such as forest monitoring, disaster management, space exploration, factory automation, secure installation, border protection, battlefield surveillance, etc., and is the basis of future network "Internet of Things" (IoT)[3]. A WSN is composed of spatially scattered sensor nodes over a certain region to monitor a certain number of tasks. The sensor nodes are positioned near to the target they are monitoring. The sensors correspond with one or more central positions, normally called base station or a sink. The more modern networks are bi-directional, also enabling control of sensor activity. A distinct sensor node contains a logic unit, a process unit to execute computations, a radio unit to unite nodes to the system and a battery power unit. Due to lack of regular power source of sensor nodes, energy preservation become the major difference between WSN and other wireless networks, and draw most attention in algorithms and protocols development.

A great challenge in this new field is achieving a global coordinated objective while using only local information [4]–[17]. Maintaining connectivity has become one of the most important and challenging task in WSN control [18]–[29]. When designing, deploying and exploiting a WSN, it is desired to maximize the number of nodes that are able to transmit or receive data to or from the base station (BS), considering the BS as an element in the network acting as a gateway between the WSN and any other data networks. Unfortunately, the number of connected nodes in a WSN (nodes that are able to send or receive data to or from the BS) would decrease over time as nodes relaying messages crash or fail due to hardware failures, battery discharge, software bugs and so forth.

There are several approaches and strategies that will minimize such risks and will improve the WSN connectivity. Most of them are high resource consuming as nodes are checking their availability of reaching a BS from time to time. The network is flooded with control messages that impact negatively on the performance and the energy efficiency of all the nodes in the path. One of the approaches provides a better trade-off between the network connectivity and the resource consumption, which will be adopted in our project as a starting point, on top of which a fuzzy logic based connectivity control will be carried out to further improve the performance. In order to better understand the effect of fuzzy logic function on this issue, different functions will be implemented and compared, and the one with the best result will be shown to guide future direction in the field.

Theoretical bases and literature review

Cooperative control of multi-agent systems is a very active research area of control theory. In the past few years problems such as flocking, consensus, coverage and pattern formation have been studied. The study is generally focused on the development of distributed control laws in order to reach a global objective [1]–[7]. One interesting problem recently analyzed regards the connectivity maintenance of the distance-dependent graph of the network. In such a graph, known also as R-disk graph [7], there is an edge between two nodes if their Euclidean distance is less than or equal to a pre specified number R. In such a graph, known also as R-disk graph [30], there is an edge between two nodes if their Euclidean distance is less than or equal to a pre specified number R. The difficulty in connectivity maintenance stems from the fact that connectivity is an inherently global property and a complicated function of the motion of the nodes. Other attempts to model changes in topology, such as [31], ignore the dependence of switching on motion. Several attempts have been made in the wireless networking literature to follow local rules that guarantee connectivity. One example is the “sector rule” which guarantees connectivity of the R-disk graph on the plane if each agent has at least one neighbor in every sector of 120 degrees [32]. Another interesting solution to the connectivity problem is given by the circumcenter algorithm, which increases gradually the degree of each agent and constraints the motion of the agents to avoid the lost of previously present connections [30]. Many authors in the control theory literature have also made progress on this problem [33]–[36].

The strategy in our project aims at keeping constant the node degree (ND) of a node, its number of neighbors. The node degree depends on the WSN deployment (regular, random, etc.), the area to be covered and the number of nodes. So, the desired node degree will be calculated for the specific WSN to be deployed and that value will become the target of the self-adaptive system. The average degree should not be too large because a large degree typically implies that a node has to communicate with other distant nodes directly. This increases interference and collision, and would waste energy. The average degree should not be too small either because that tends to increase the overall network energy consumption as longer paths have to be taken. So we believe the average node degree is an important performance metric for multihop wireless network topology. Intuitively, if a node has a higher degree (indicates more neighbors), it is more likely that there are at least one path for it to transmit data to the destination. If all nodes are randomly and uniformly deployed, the probabilistic approach to analyze the relation between the node degree and the network connectivity is fully described in [37].

The self-adaptive system presented here aims to control the communication range of each node to manage its degree, in order to recover the link when its neighbors fail. Whenever a node's neighbor fails, the communication range of that node is increased to replace the failing neighbor. Therefore the node's energy consumption is likely to increase. If the desired node degree, whose value is estimated before the nodes deployment as mentioned above, is kept constant all the time, the battery might become exhausted too short. Thus, the desired node degree has to be adjusted in run time taking into account the battery level and the desired lifetime of autonomous nodes. Note that directly changing communication range usually is not feasible, instead the transmission power is the parameter can be controlled in real sensor nodes. In this paper, we employed the transmission power model in the literature [38], in which the transmission power is linear function of square of the communication range.

The basic idea of the control system is that if the node degree is higher than the expected node degree, then the communication range has to be decreased; if the node degree is lower than the expected node degree, then the communication range has to be increased. The desired node degree will depend on the energy of the node. How fast and how long the communication range changes, is decided by the controller, e.g. fuzzy logic based controller.

First we need to understand what exactly is fuzzy logic. Fuzzy logic is a form of many-valued logic which deals with reasoning that is approximate rather than fixed and exact. Fuzzy logic has two different meanings. In a narrow sense, fuzzy logic is a logical system, which is an extension of multivalued logic. However, in a wider sense fuzzy logic (FL) is almost synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with unsharp boundaries in which membership is a matter of degree. In this perspective, fuzzy logic in its narrow sense is a branch of FL. Even in its more narrow definition, fuzzy logic differs both in concept and substance from traditional multi valued logical systems[39].

Another basic concept in FL, which plays a central role in most of its applications, is that of a fuzzy if-then rule or, simply, fuzzy rule. Although rule-based systems have a long history of use in Artificial Intelligence (AI), what is missing in such systems is a mechanism for dealing with fuzzy consequents and fuzzy antecedents. In fuzzy logic, this mechanism is provided by the calculus of fuzzy rules. The calculus of fuzzy rules serves as a basis for what might be called the Fuzzy Dependency and Command Language (FDCL). Although FDCL is not used explicitly in the toolbox, it is effectively one of its principal constituents. In most of the applications of fuzzy logic, a fuzzy logic solution is, in reality, a translation of a human solution into FDCL.

First fuzzy logic composed of Membership Functions(MF) which is a curve that defines how each point in the input space is mapped to a membership value (or degrees of membership) between 0 and 1. For example, when we define somebody is tall, we don’t have a clear boundary between above which height is considered tall. We only have a rough idea of the definition of tall. In this case we can only use curve instead of linear lines to represent the tallness. As described in Fig 1 and 2, it’s obviously more sensible to define tall in the curve format. When someone is 5 ft 9 inches, we can say he is tall to the degree of 0.7. And we can map these values accordingly on to the curve function.

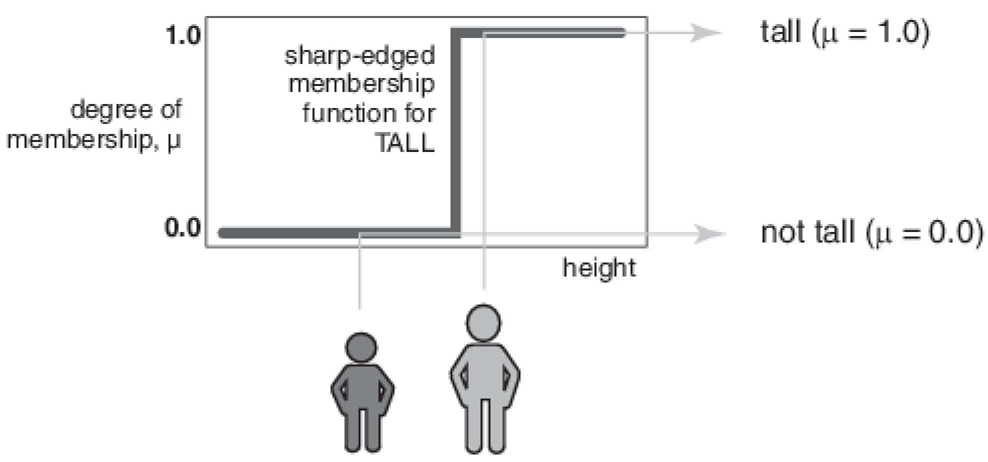


Figure1. Linear function to define tallness

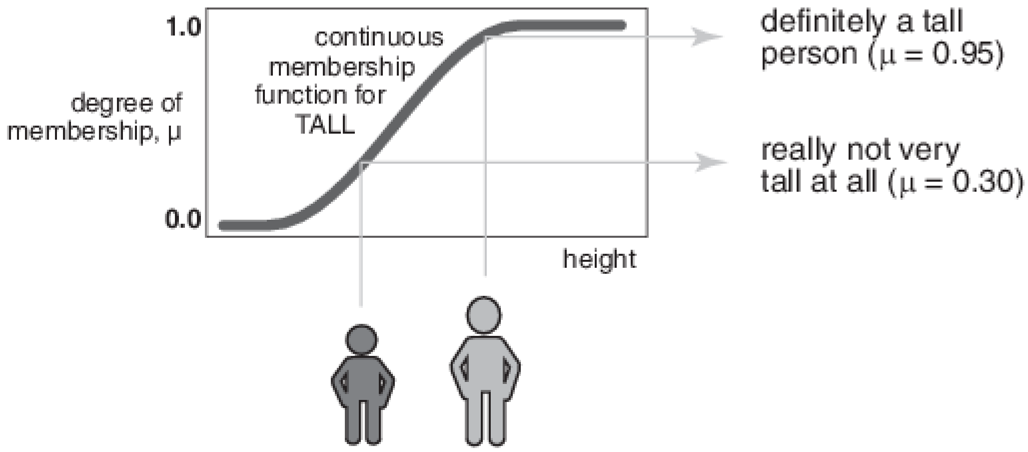


Figure2. curve to define tallness

The second component of fuzzy logic is the If-Then rules. Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. These if-then rule statements are used to formulate the conditional statements that comprise fuzzy logic.

The simple format of fuzzy if-then rule is :

- if x is A then y is B.

where A and B are linguistic values defined by fuzzy sets on the ranges X and Y. Taken our previous example on the tallness, we can define a if-then rule as “If a man is 5 ft. 9 inches, then he is tall with a degree of 0.7. In general, the input to an if-then rule is the current value for the input variable and the output is an entire fuzzy set[40] .

The last component is the fuzzy inference. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic[41] . The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves all of the pieces that are described in Membership Functions, Logical Operations, and If-Then rules. We take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The output is a fuzzy degree of membership in the qualifying linguistic set which always are the intervals between 0 and 1.We also need to aggregate all outputs in order to make decisions based on the results. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Lastly is the defuzzify process. The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number which is the result we want to derive from the fuzzy inference process.

Hypothesis

There are several approaches and strategies that will minimize the risks of message delaying fails, and thus improve WSN connectivity. However most of them will require high power consumption since each node is actively checking others' availability of reaching the BS. We aim to tackle both problems of poor connectivity and also to save power by using the fuzzy logic controller on each node to monitor the node's own parameters, without flooding WSN with monitoring messages. We use control loops based on fuzzy logic to enable each node to adjust automatically the communication range according to a desired node degree and residual energy.

Methodology

Generate/collect data

a. The base station is positioned at the center of the 100 * 100 area
b. Generate 30 random sensor nodes in the area and initialize every node with a random communication range within a given range (10 to 30).
c. Other strategies to initialize the network: 1). deploy the nodes semi-randomly by fixing each node within a certain area; 2). initialize the communication range with a constant
d. In each cycle, update the status of every node and then transmit a packet from each node to base station(success or fail) and count

How to solve problem

Algorithm

The algorithm contains three loops: main loop, primary loop, secondary loop (Figure 1)

i. **Main loop**: update the whole system

- Update the status of every node
- Calculate the paths from sensor nodes to base station (undirected and weighted graph)
- Monitor the packet transmission from sensor nodes to base station
- Do analysis of the whole system and display and update the result.

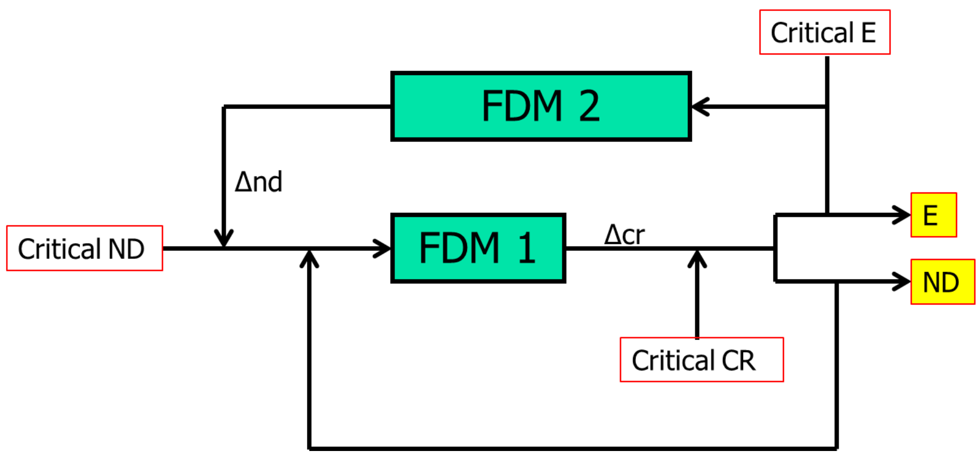


Figure 1. Flow chart of fuzzy-logic based control system

ii. **Primary loop**: update each node of its communication range

- <u>Inputs</u>: current energy level and node degree of a node, and a set of predefined parameters including:

  initial value of the communication range ($\overline{CR_0}$); desired value of the node degree when the battery has a critical energy level ($\overline{ND}$); critical energy level ($\overline{E_{cr}}$); communication range variation rate ($\overline{\Delta cr}$); node degree variation rate ($\overline{\Delta nd}$); minimum and maximum value of the communication range (CR_{min} and CR_{max}).

- <u>Loop</u>: input modification and normalization, and fuzzy-logic based function of decision making

(1) Membership functions: for comparison, two configurations, trapezoidal-triangle mixed shape (Figure 2) and triangle-shape (Figure 3) are used to evaluate the normalized changes (e) of node degree

$$triangle-shape \qquad \mu(x) = \begin{cases} 0, & x \le a; x \ge c \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \end{cases}$$

$$trapezoidal-shape \qquad \mu(x) = \begin{cases} 0, & x \le a; x \ge d \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \end{cases}$$

(2) Input space is partitioned in three regions: negative values(NV), zero values(ZV) and positive values(PV), and their corresponding output variable (u) is:

$$\Delta u = \begin{cases} f_{NC} = -1, & e \in NV \\ f_{NC} = 0, & e \in ZV \\ f_{NC} = 1, & e \in PV \end{cases}$$

region	MF	a	b	c	d	boundaries
NV	$\mu_{NV}(e)$	-4	-2	-0.5	0	$e < 0$
ZV	$\mu_{ZV}(e)$	-0.5	0	0.5		$-0.5 \le e \le 0.5$
PV	$\mu_{PV}(e)$	0	0.5	2	4	$e > 0$

Table 1. Parameters of membership functions with mixed shape

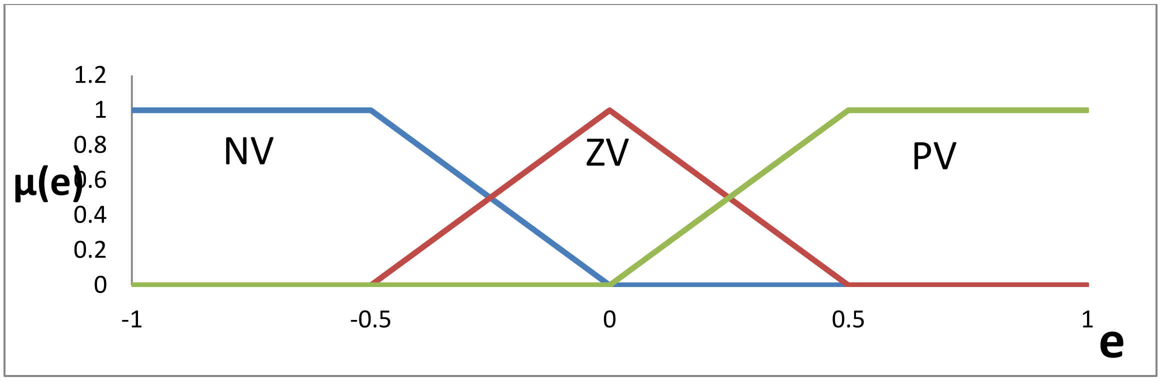


Figure 2. Membership functions of trapezoidal-triangle mixed shape

region	MF	a	b	c	boundaries
NV	$\mu_{NV}(e)$	-2	-1	0	$e < 0$
ZV	$\mu_{ZV}(e)$	-1	0	1	$-1 \le e \le 1$
PV	$\mu_{PV}(e)$	0	1	2	$e > 0$

Table 2. Parameters of membership functions with triangle shape

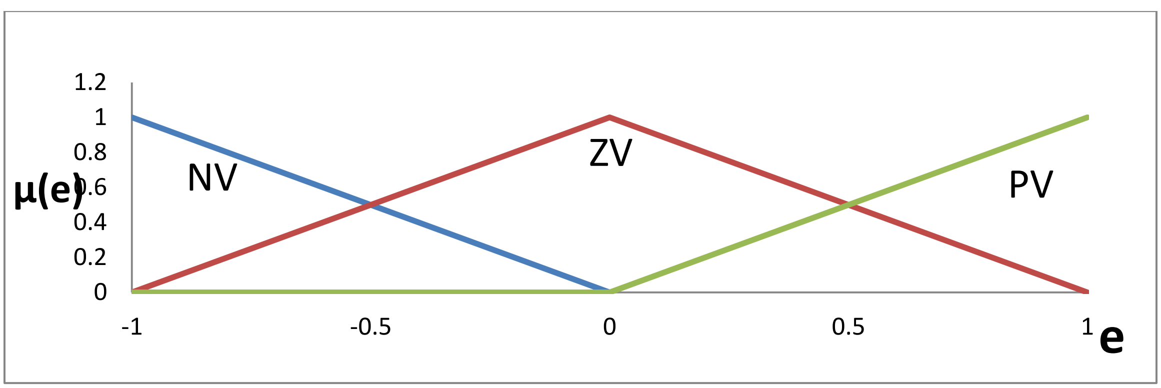


Figure 3. Membership functions of triangle shape

- Outputs: normalized communication range variation factor which can be used to calculate the updated communication range and update energy level after packet transmission

iii. Secondary loop: update the desired node degree of each node using its battery level. Same fuzzy-logic membership functions as the primary loop are used.

Language

We will implement the program in Java. The simulation and data analysis will be visualized using Java GUI tools.

Tool

JFreeChart (http://www.jfree.org/jfreechart/)

Generate output

a. Draw the nodes network connected graph against time (cycle number) to display the change in connectivity of the network.
b. Draw the shortest distance path from each sensor node to the based station to visualize the packet transmission in each cycle.
c. Plot the total remaining energy of the system against time(cycle number)
d. Record the total number of packets received at based station

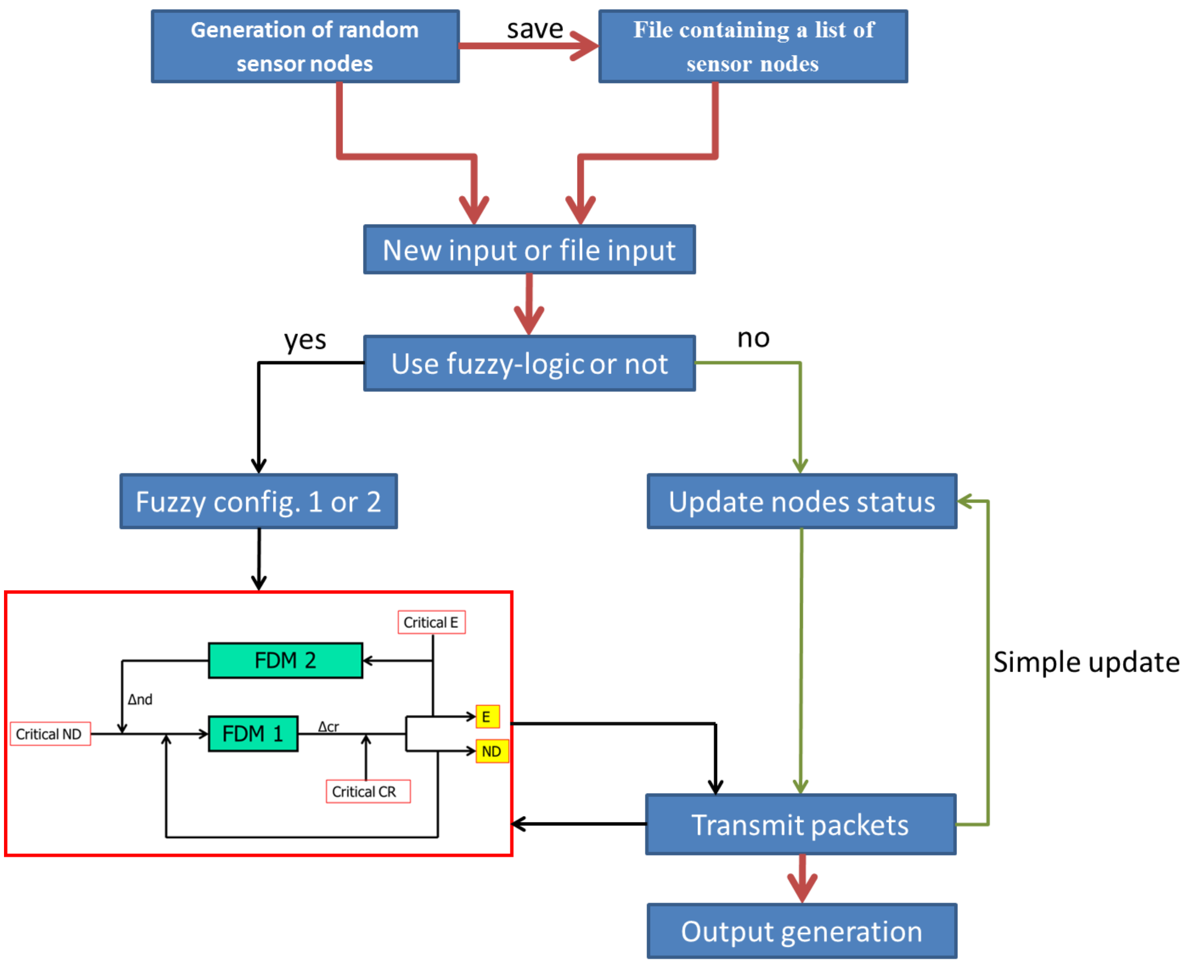
Test against hypothesis

a. Compare the network connection in each cycle and show the dynamic change of the connected sensor network and the packet transmission paths.
b. Compare the number of packets received at based station of fuzzy-logic based system with that of static non-fuzzy system to show the advantages of fuzzy-logic controlled system
c. Compare numbers of packets transmitted using different fuzzy-logic functions to show the choice of fuzzy-logic functions can affect the performance of sensor network
d. Compare the changes of total network energy between fuzzy-logic based and static systems to show the different energy consumption footprints in these two types of systems

Implementation

Code (see attachment)

Design document and flow chart



a. Generation of a list of nodes including a based station node and 30 random sensors nodes. The newly generated list is saved to a txt file so that the same input data can be used for simulations with and without fuzzy logic as well as different fuzzy-logic configurations.

b. To start the simulation, a list of nodes is either generated at will or imported from a txt file, followed by the generation of the starting sensor network graph.

c. Before simulation, options for configurations can be selected: fuzzy-logic or non-fuzzy-logic, fuzzy-logic configuration 1 or 2

d. In the main loop, at most 30 cycles of simulations are performed. In each cycle, the status of each node (communication range) is updated by fuzzy-logic function using its battery level and node degree. Then the new network graph and shortest path graph are calculated, displayed and saved as a png file. The total remaining energy and number of packets transmitted in each cycle are also saved for data analysis.
e. Shortest path graphs are calculated as single-drain shortest-path graphs
f. After the loop, the total number of packets transmitted in the simulation is displayed in console, and a graph plotting the total remaining energy in each cycle is shown.

Data analysis

Output generation:

To study the influence of fuzzy-logic on the wireless sensor network performance, we generate multiple sets of input data. And we compare the performance of WSN with three different configurations: fuzzy-logic configuration 1, fuzzy-logic configuration 2 and non-fuzzy-logic configuration. The specific parameters for comparison are: total packet received at base station and total remaining energy of the network of each cycle. In general, based on the number of packets transmitted, the data sets can be divided into three groups:

i. Cases with fairly large improvement
ii. Cases with a small improvement
iii. Cases with no improvement

Case 1 with fairly large improvement

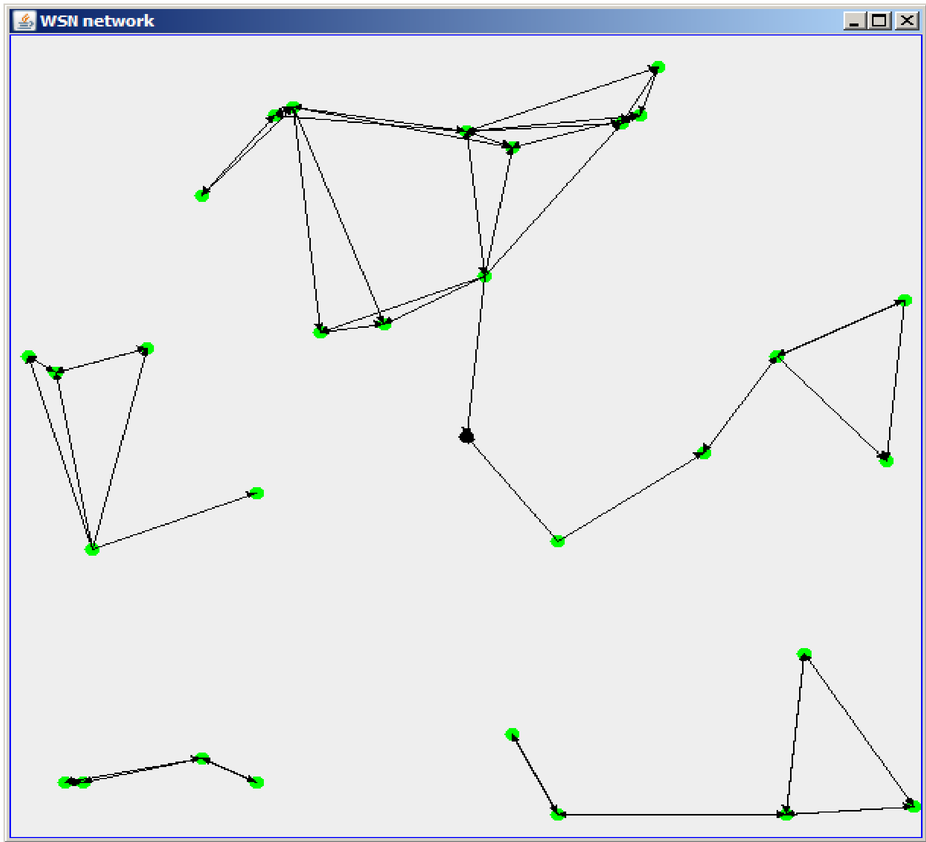
List of nodes

Node Index	x	y	CR
0 (base station)	50	50	0
1	98	33	24
2	50	12	25
3	6	93	19
4	29	10	25
5	41	36	11
6	96	53	10

7	85	97	28
8	15	39	13
9	84	40	26
10	21	90	21
11	55	87	19
12	71	4	18
13	34	37	19
14	31	9	29
15	67	11	12
16	21	20	15
17	55	14	11
18	52	30	25
19	76	52	15
20	27	93	18
21	99	96	26
22	5	42	22
23	9	64	27
24	69	10	25
25	27	57	17
26	2	40	12
27	8	93	14
28	60	63	20
29	60	97	29
30	87	77	25

WSN network graph before simulation

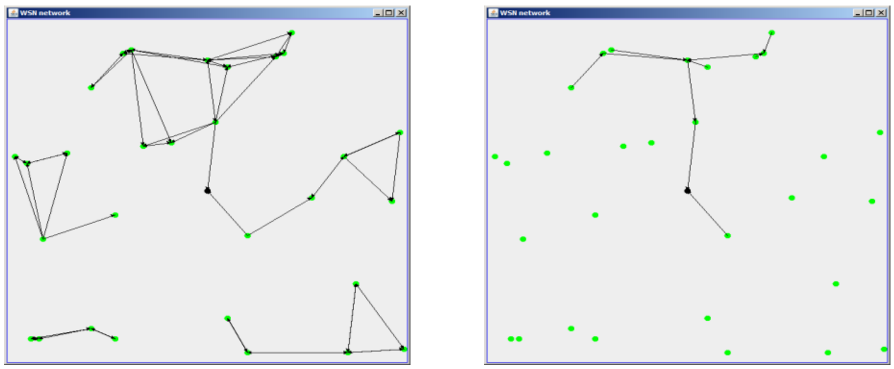
In the graph, the green dots are sensor nodes and black dot in the center is the base station. The arrowed lines indicate a directed connection from one node to another.



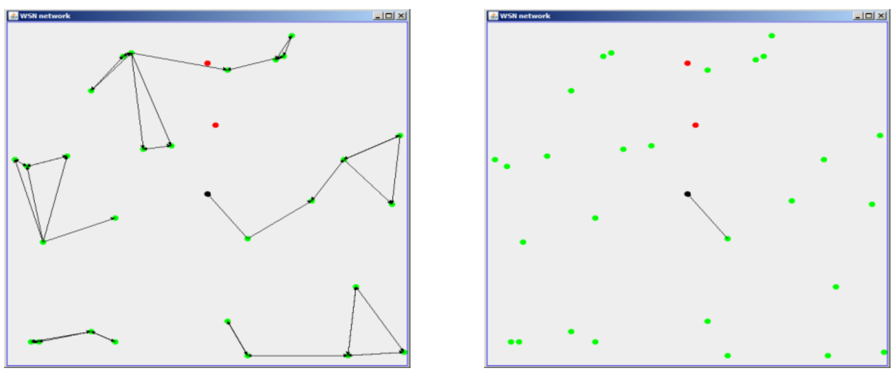
Simulation without fuzzy-logic based control

The left pictures are connected graphs of the network and right pictures are calculated shortest path graph. As we can see from the graphs, the basic structure of the WSN connection graph (shown in left pics) does not change. The only change is that as time goes by, some of the connections are broken due to the depletion of energy of some nodes (red dots). Meanwhile, the calculated shorted paths exhibit similar change. Because communication range of every node is fixed, no new connection can be formed after some of the nodes lose power.

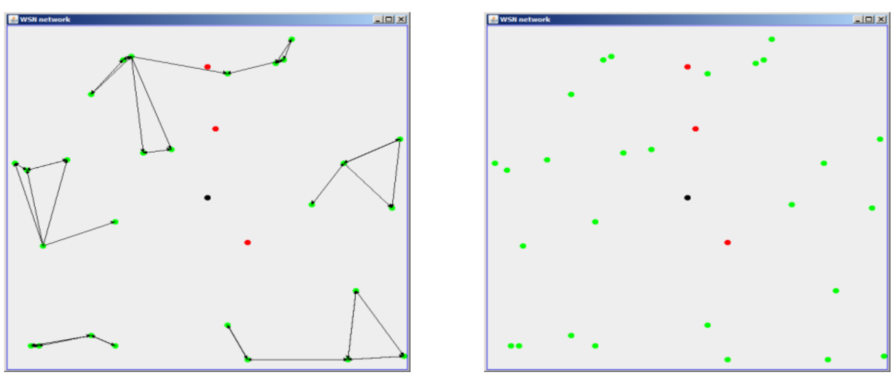
Cycle #1



Cycle #15



Cycle #26

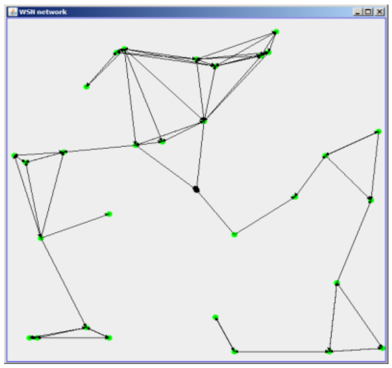


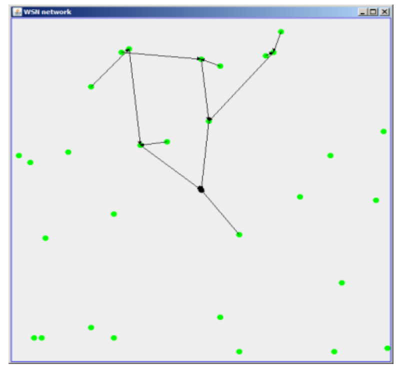
Dynamic changes of WSN network and shortest path graphs

Different from the behavior of the static WSN network, both the left and right graphs changes as simulation continues. This shows that the WSN controlled by fuzzy logic can dynamically adjust its connection and the shortest path graph, due to the fact that every node can change its communication range through fuzzy-logic feedback system using its current energy level and node degree.

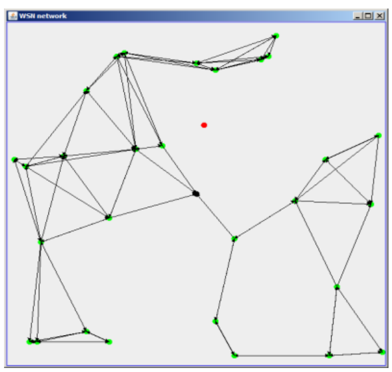
- Fuzzy logic configuration 1

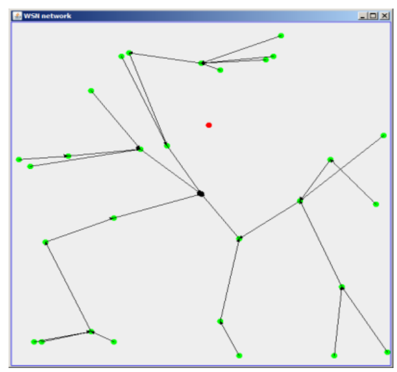
Cycle #1



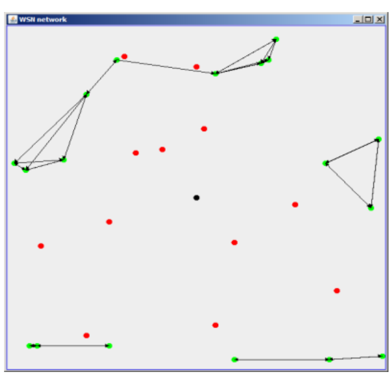


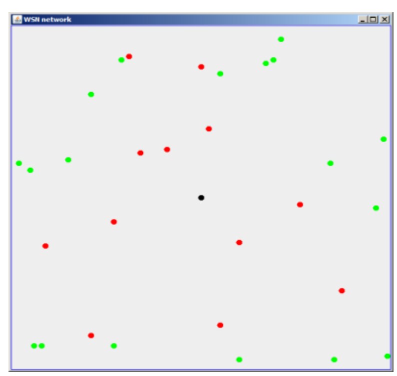
Cycle #4





Cycle #7

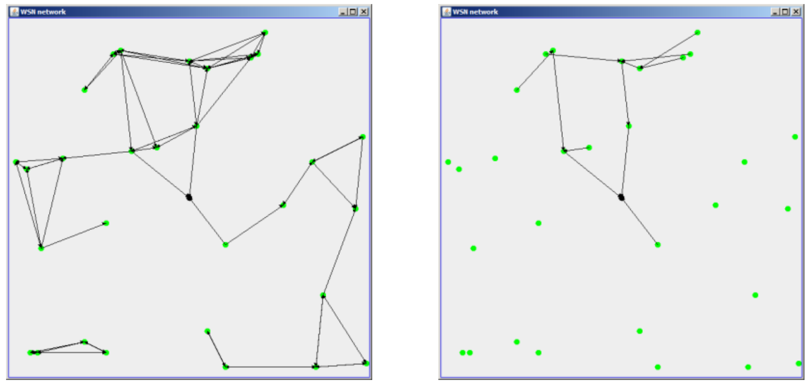




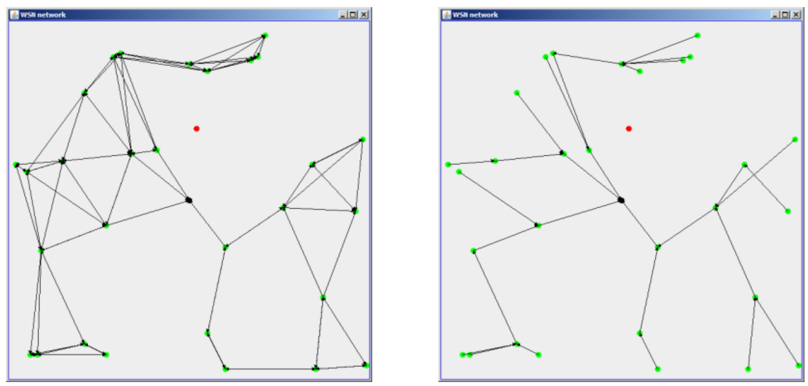
- Fuzzy logic configuration 2

The simulation result is almost the same as that of fuzzy logic configuration 1.

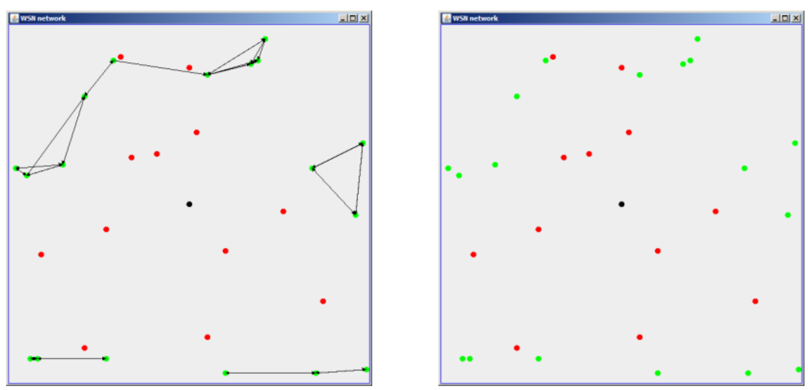
Cycle #1



Cycle #4



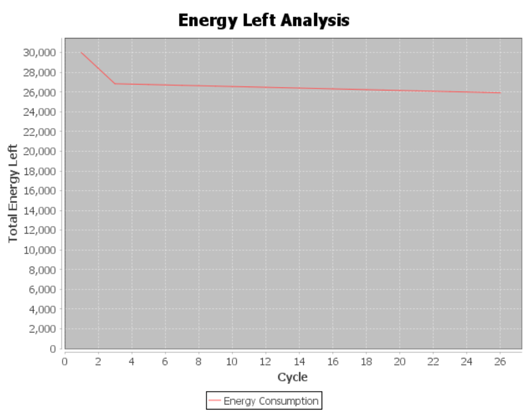
Cycle #7



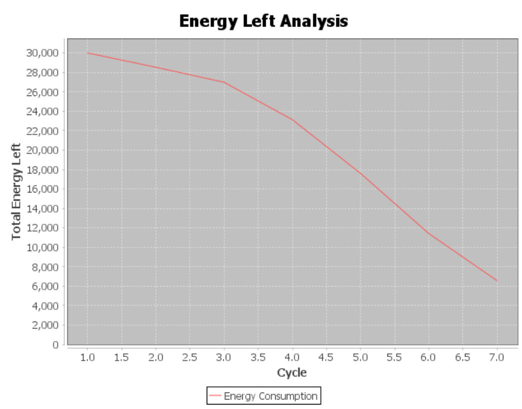
Footprint of total WSN energy

As we can see from the graphs, in simulation without fuzzy-logic, the total energy of the WSN network decreases by a small percentage followed by a plateau. The shape of the curve shows after the initial packet transmission, no signification further transmission is observed, which means in real WSN system, the data collection at base station is stopped and WSN loses its function. In comparison, the system with fuzzy logic control keeps consuming energy and transmitting packets to base station even when a significant percentage of the energy is consumed. So this type of WSN is able to collect more data and make better use of its resources.

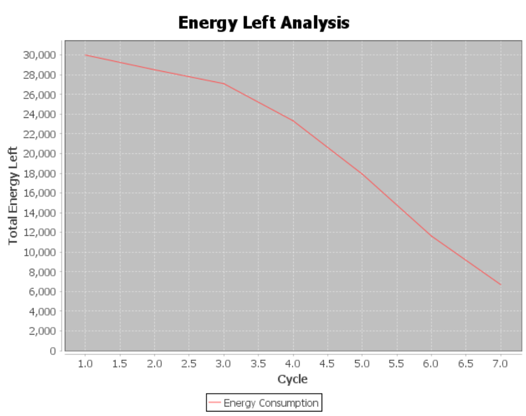
Non-fuzzy logic



Fuzzy logic config. 1



Fuzzy logic config. 2



Total number of packets transmitted in simulation

	Non-fuzzy logic	Fuzzy-logic config. 1	Fuzzy-logic config. 2
Number of packets	43	120	120

Case 2 with a small improvement

The chance of getting this type of cases in random generation of nodes are lower than the case 1. In this case, the improvement of packet transmission is small, although the fuzzy-logic controlled system still shows dynamic changing in connection graph and shortest path graph.

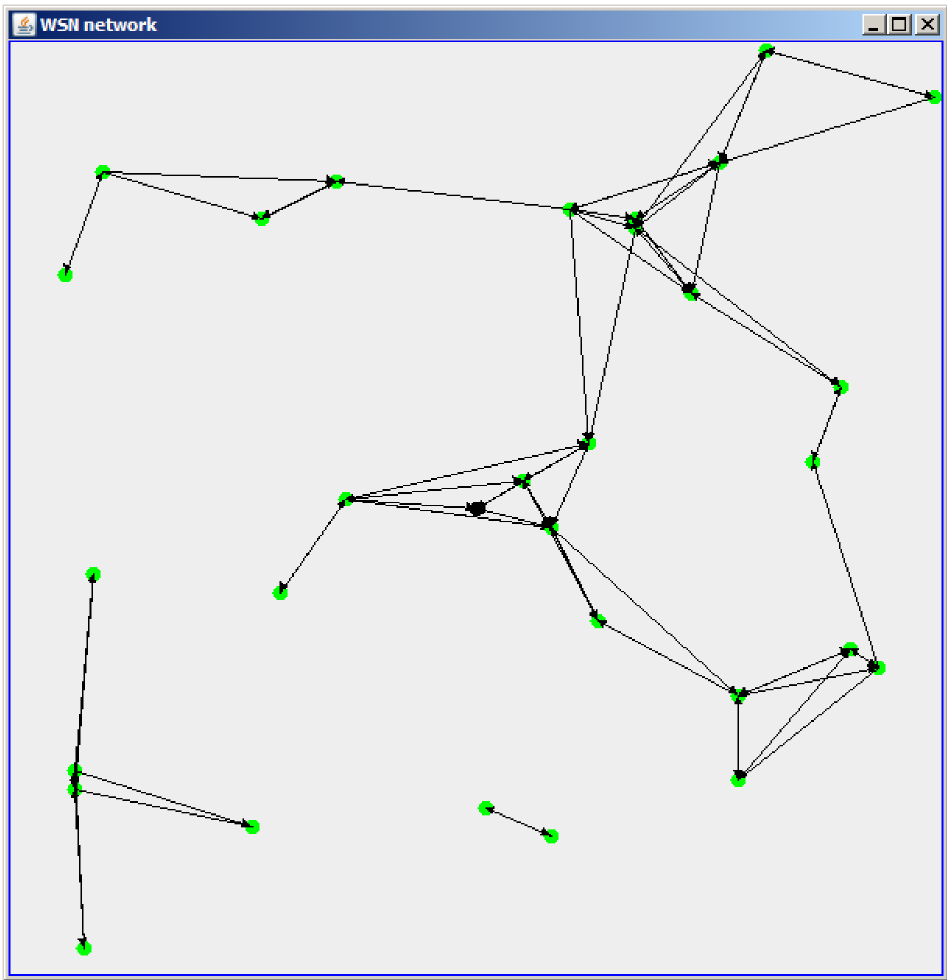
List of nodes

Node Index	x	y	CR
0 (base station)	50	50	0
1	67	20	29
2	73	27	12
3	93	67	28
4	90	65	13
5	51	82	23
6	78	70	17
7	10	14	27
8	35	15	19
9	62	43	17
10	89	37	23
11	58	52	28
12	55	47	23
13	81	1	21
14	9	57	12
15	76	13	19
16	58	85	10
17	99	6	27
18	26	84	13
19	67	19	21
20	63	62	13
21	86	45	18
22	36	49	28
23	27	19	15
24	8	97	22
25	29	59	15
26	7	80	24
27	7	78	26
28	6	25	12

29	78	79	19
30	60	18	27

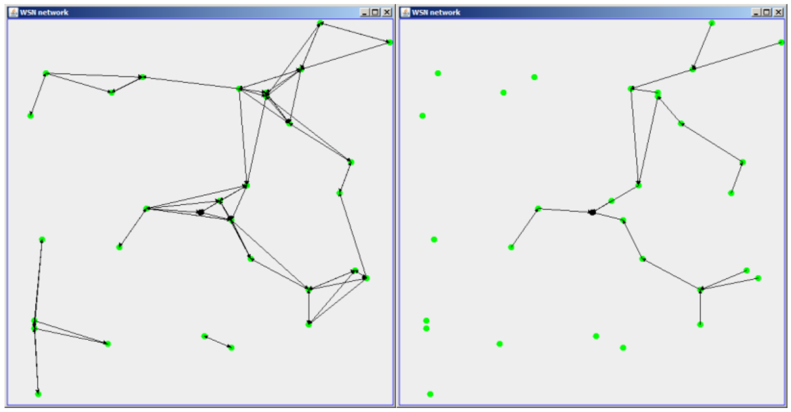
WSN network graph before simulation

In the graph, the green dots are sensor nodes and black dot in the center is the base station. The arrowed lines indicate a directed connection from one node to another.

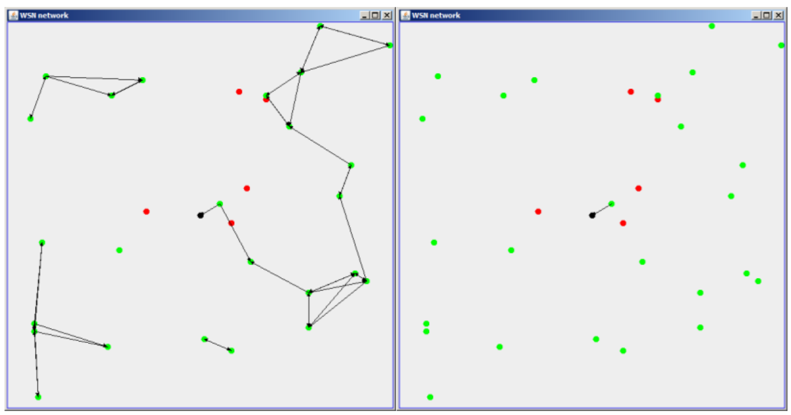


Simulation without fuzzy-logic based control

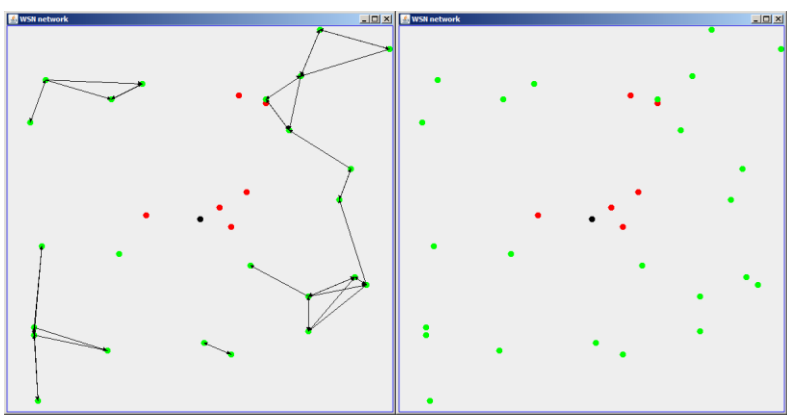
Cycle #1



Cycle #11



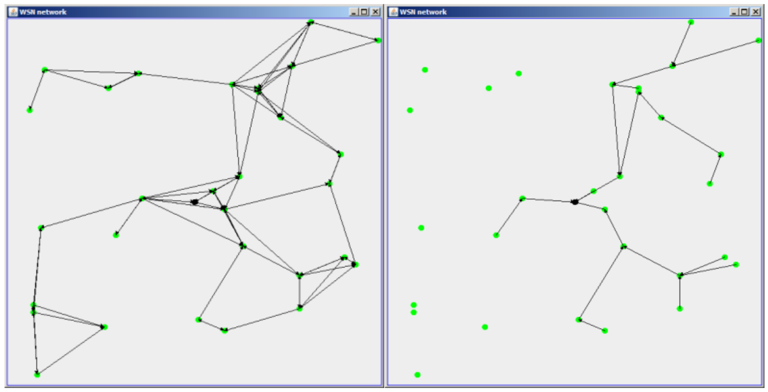
Cycle #21



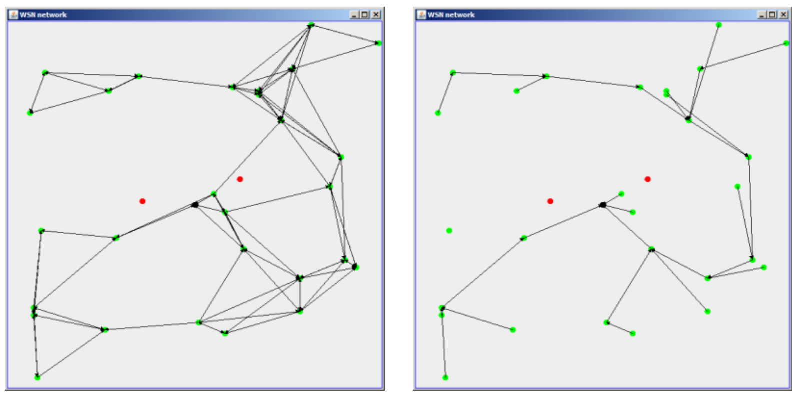
Dynamic changes of WSN network and shortest path graphs

- Fuzzy logic configuration 1

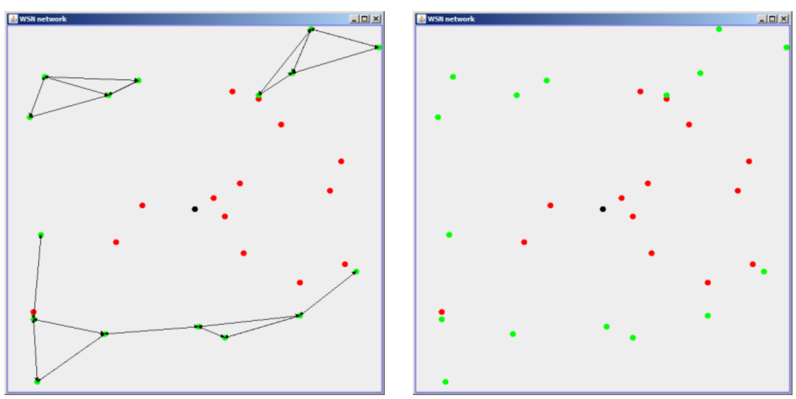
Cycle #1



Cycle #4



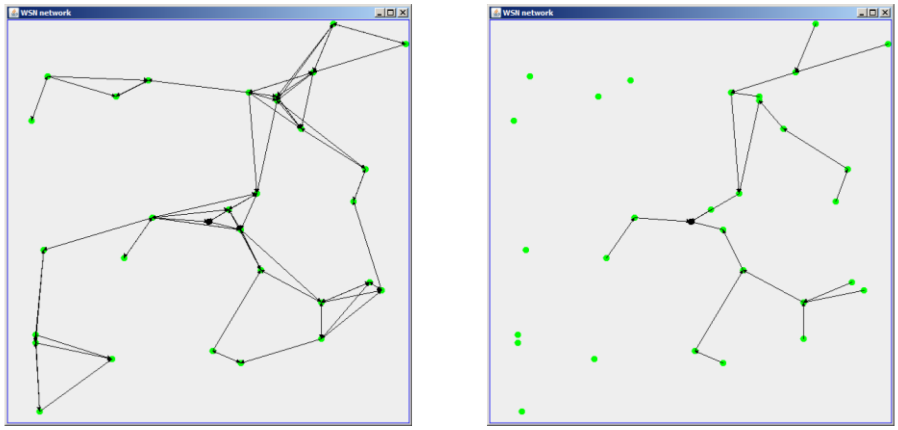
Cycle #7



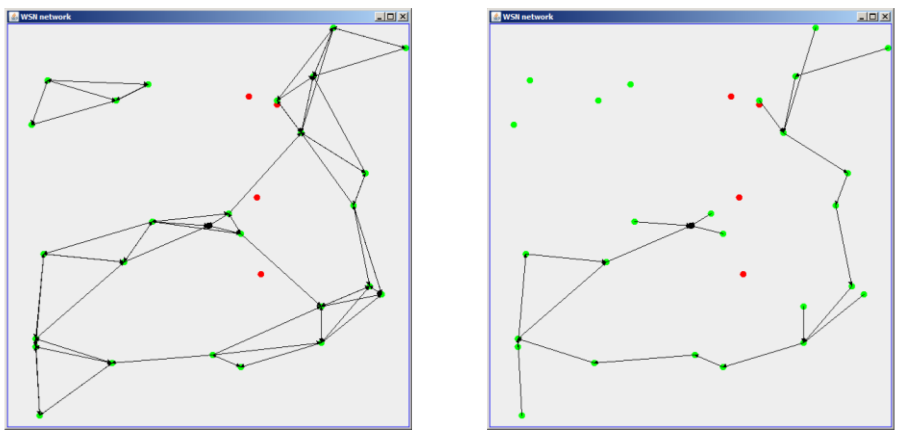
- Fuzzy logic configuration 2

The simulation result shows dynamic change of graph but is different from that of fuzzy logic configuration 1.

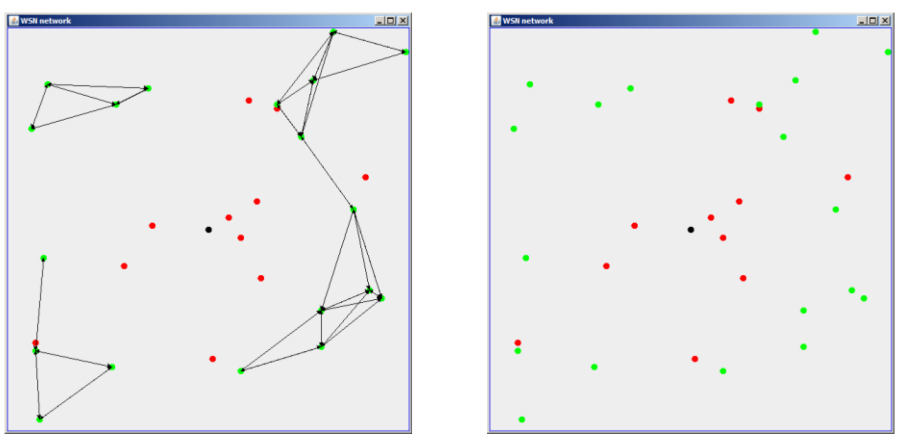
Cycle #1



Cycle #5



Cycle #10

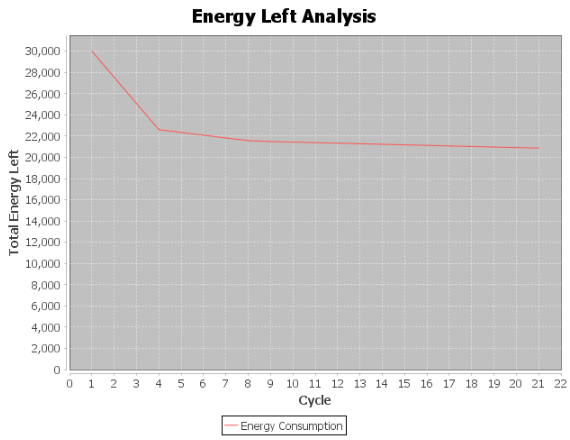


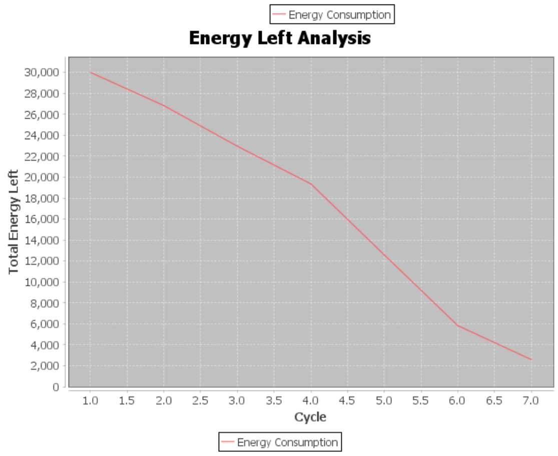
Footprint of total WSN energy

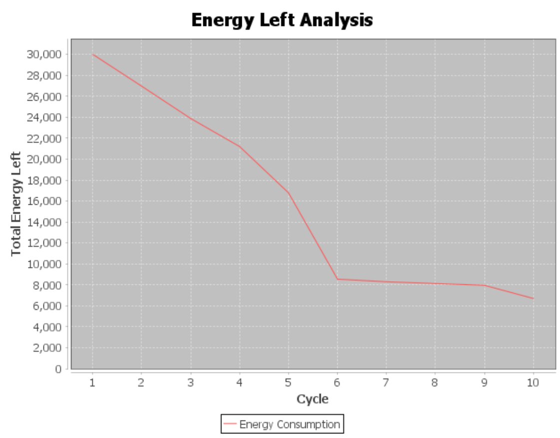
Behavior is similar to that in case 1

Non-fuzzy logic

Fuzzy logic config. 1







Fuzzy logic config. 2

Total number of packets transmitted in simulation

	Non-fuzzy logic	Fuzzy-logic config. 1	Fuzzy-logic config. 2
Number of packets	87	133	120

Case 3 with no improvement

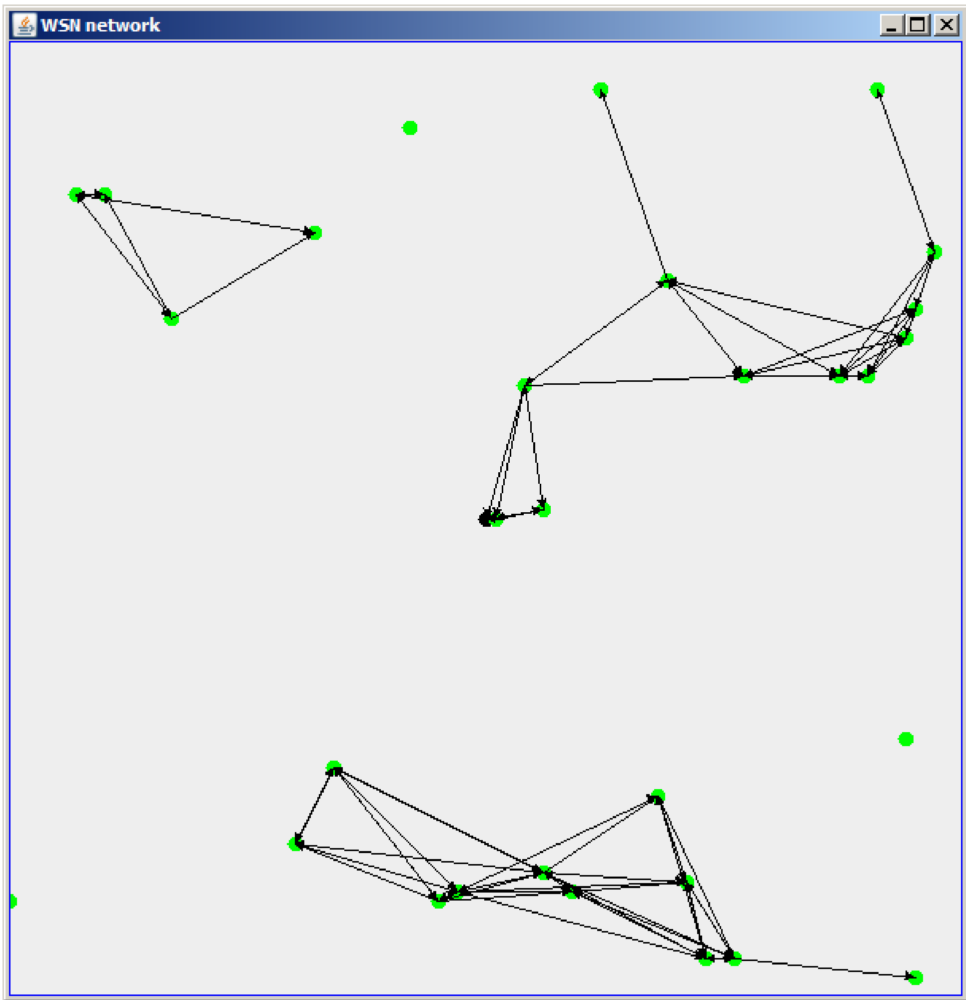
The chance of getting this type of cases in random generation of nodes are lower than the case 1 and 2. In this case, the improvement of packet transmission is not obvious, and the fuzzy-logic controlled system doesn't show obvious dynamic changing in connection graph and shortest path graph.

List of nodes

Node Index	x	y	CR
0 (base station)	50	50	0
1	94	73	21
2	47	89	29
3	95	98	19
4	30	84	15
5	51	50	13
6	90	35	11
7	45	90	25
8	42	9	11
9	0	90	17
10	94	31	26
11	73	96	20
12	7	16	29
13	54	36	25
14	56	49	16
15	10	16	18
16	62	5	14
17	97	22	22
18	76	96	24
19	91	5	19
20	77	35	21
21	68	79	10
22	95	28	11
23	69	25	22
24	32	20	13
25	59	89	12
26	17	29	20
27	87	35	13
28	56	87	29
29	71	88	15
30	34	76	25

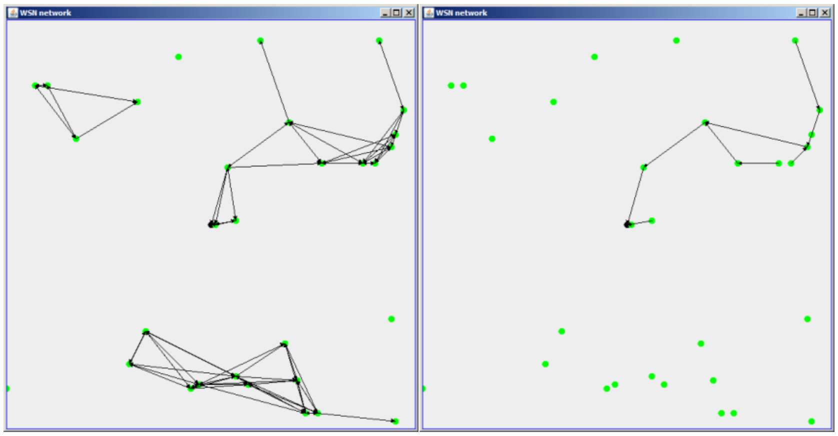
WSN network graph before simulation

In the graph, the green dots are sensor nodes and black dot in the center is the base station. The arrowed lines indicate a directed connection from one node to another.

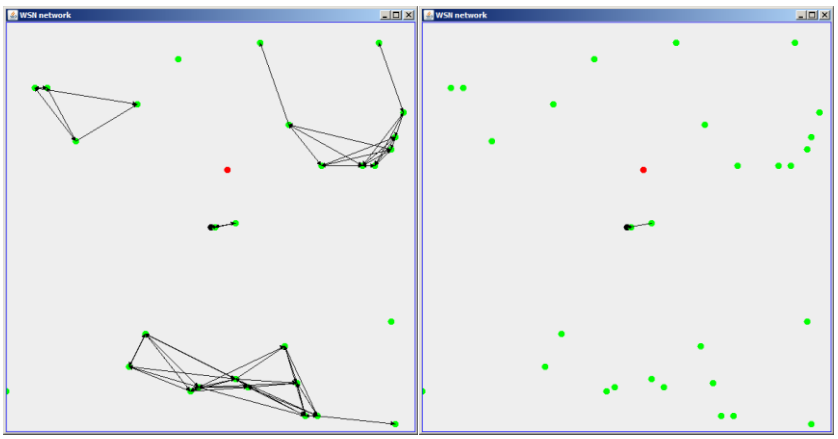


Simulation without fuzzy-logic based control

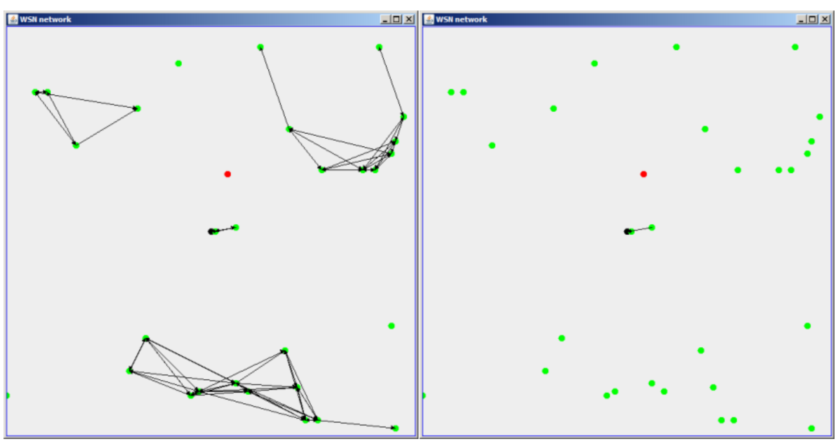
Cycle #1



Cycle #15



Cycle #30

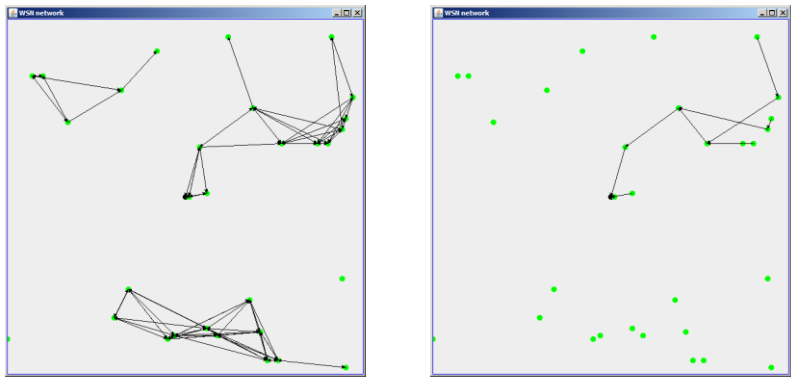


Dynamic changes of WSN network and shortest path graphs

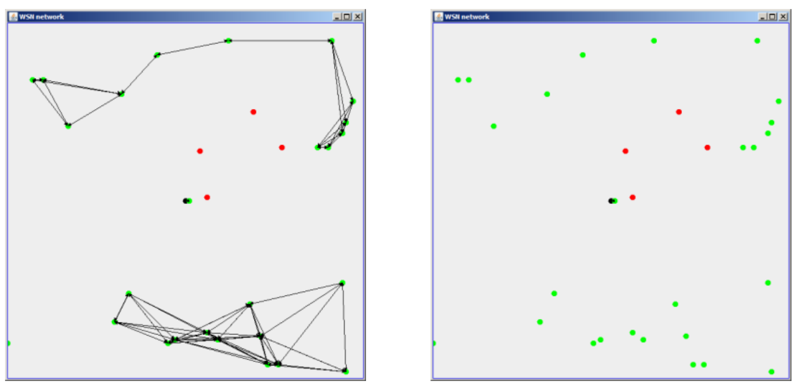
- Fuzzy logic configuration 1

  The shortest path is similar to that of non-fuzzy logic system. As the energy of the node consumes, the shortest path graph does not change much compared with the previous 2 cases which mean this sensor nodes system has flawed initial state which restricts its dynamic properties.

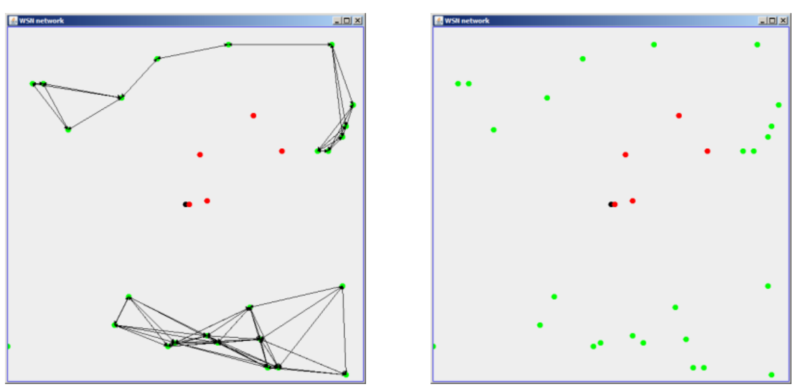
Cycle #1



Cycle #8



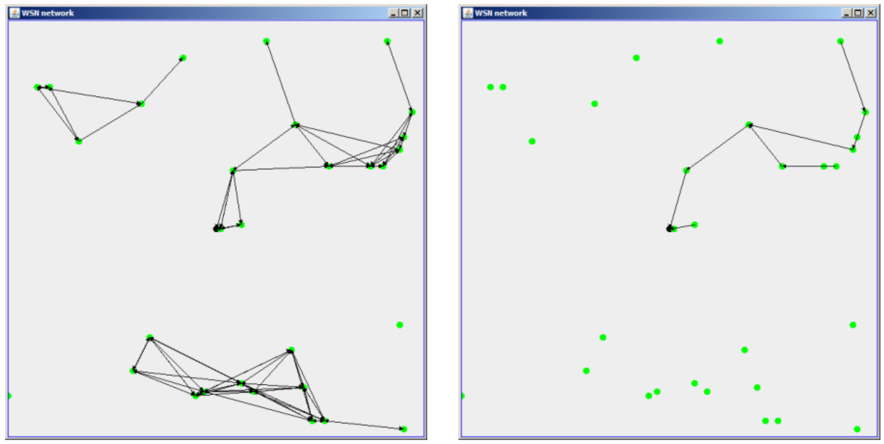
Cycle #16



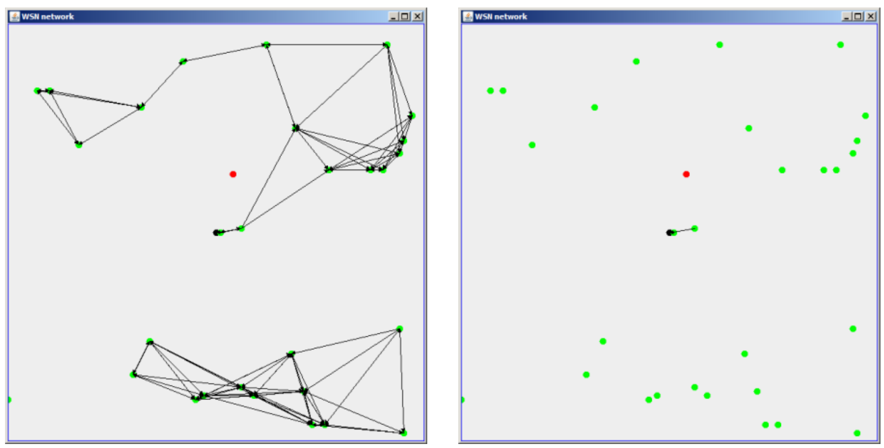
➢ Fuzzy logic configuration 2

The simulation result is similar to that of fuzzy logic configuration 1.

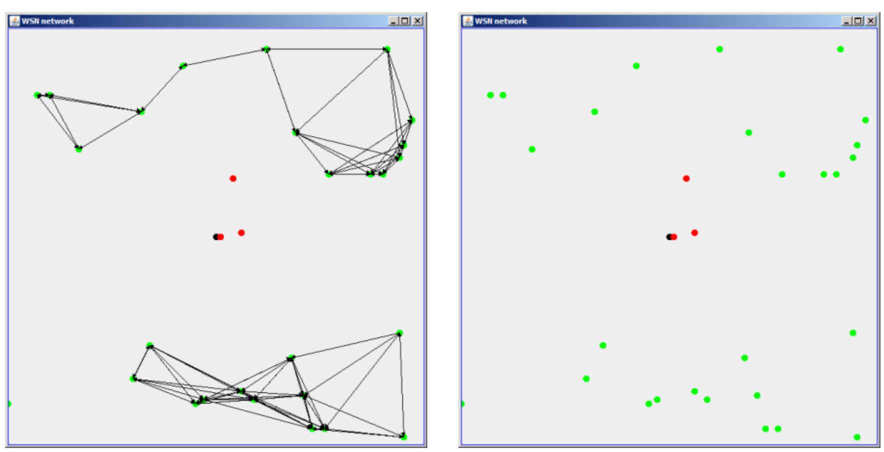
Cycle #1



Cycle #8



Cycle #16

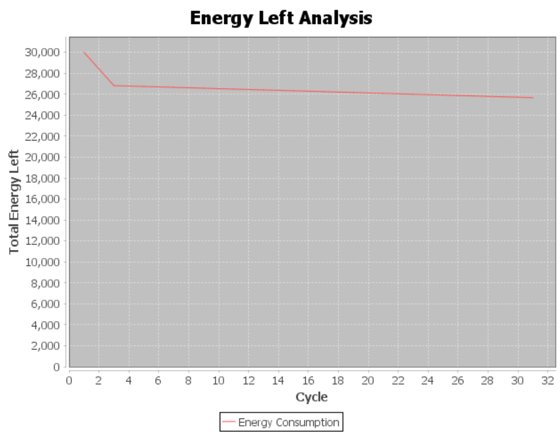


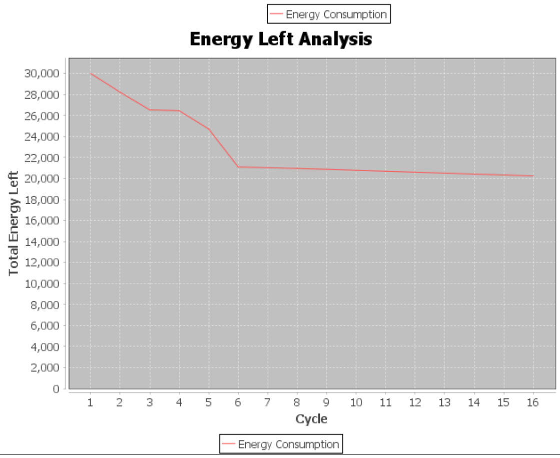
Footprint of total WSN energy

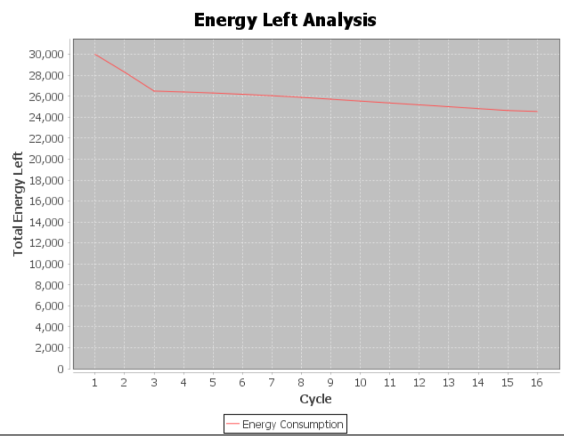
All three configurations show similar curve of energy in each cycle. At the end of simulation, all three still have a large percentage of energy left in the system, which means the WSN network does not functions well . And the data collected in the course is very limited.

Non-fuzzy logic

Fuzzy logic config. 1







Fuzzy logic config. 2

Total number of packets transmitted in simulation

	Non-fuzzy logic	Fuzzy-logic config. 1	Fuzzy-logic config. 2
Number of packets	60	47	78

i. In above special case, the improvement based on fuzzy logic is not significant. However there are a few observations that we made:

- In the connectivity graph, one critical node in the topology has run out of power. The network without fuzzy logic will immediately get disconnected, no more packets are sent to the Base Station anymore. However, for the network with fuzzy logic, the communication range of nodes automatically get adjusted to increase the node degree. As a result, new path has been identified and new connection has been established.
- In terms of the shortest path tree, once the critical node run out of power, the shortest path will be destroyed because of the disconnect. For the network with fuzzy logic, since connectivity graph has been dynamically established, new shorted path has been created. To some example, the new shorted path performs even better than the old shorted path.
- For the energy left analysis, because of the disconnect, the network without fuzzy logic will stop consuming power and the total power displays a horizontal line with no reduction. This proves to be very inefficient since most of the nodes are still fully charged. This demonstrates that the power utilization is highly dependent on the topology of the network, which in turn suggests that the topology can become a bottleneck for the entire network. However one of the most important feature of Wireless Sensor Network is the randomly distributed nodes. In contrast, the network with fuzzy logic shows better utilization of the node power. Even though in the chart the curve becomes flat as time goes by, but still more power are consumed before that which means more nodes served their job.

Fuzzy logic control has a positive impact on WSN packet transmission

In order to measure the network performance, the total number of packets received at base station during the whole simulation has been measured. For each fuzzy control function, 100 different network topologies have been simulated. The nodes for each network are randomly deployed and the simulation has been executed three times: without the fuzzy control-based algorithm, with fuzzy control using our first fuzzy function, and with fuzzy control using our second fuzzy function. The raw data is shown in Appendix Table I where the first column is the total number of packets transmitted when no fuzzy control is used (P_{BS_NC}). This is our experimental control. The second column is the number of packets transmitted (P_{BS_WC1}) and the percentage increase when using the first fuzzy logic control function. The third column is the number of packets transmitted (P_{BS_WC2}) and the percentage increase when using the second fuzzy logic control function. The percentage increase is calculated as 100% x (P_{BS_WC} - P_{BS_NC}) / P_{BS_NC}. To further analyze the data, improvement histograms are made and shown in Figure I and II. Upon calculation, it is found that the first fuzzy logic control function has an impact of an average of 77.11% improvement in the total number of packet transmission with a standard deviation of 72.71%, whereas the second fuzzy logic control function has 75.57% mean value and 72.75% standard deviation.

After an analysis of the results, the following observations can be made: (1) in general, more packages were received at BS when the fuzzy control algorithm was applied; (2) the two fuzzy logic control function have similar impact on the system performance with the first function performing slightly better; (3) in general the self-adaptive communication range based on fuzzy control loops improves the connectivity of the network.

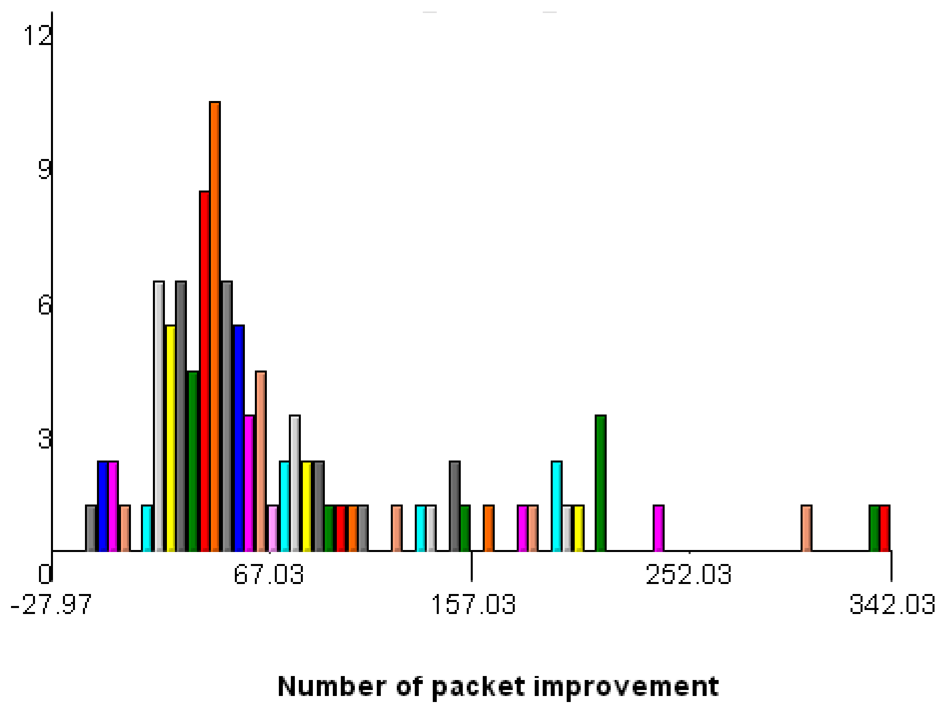


Figure I. Number of packet improvement percentage histogram of 100 randomly deployed networks with the first fuzzy logic control function.

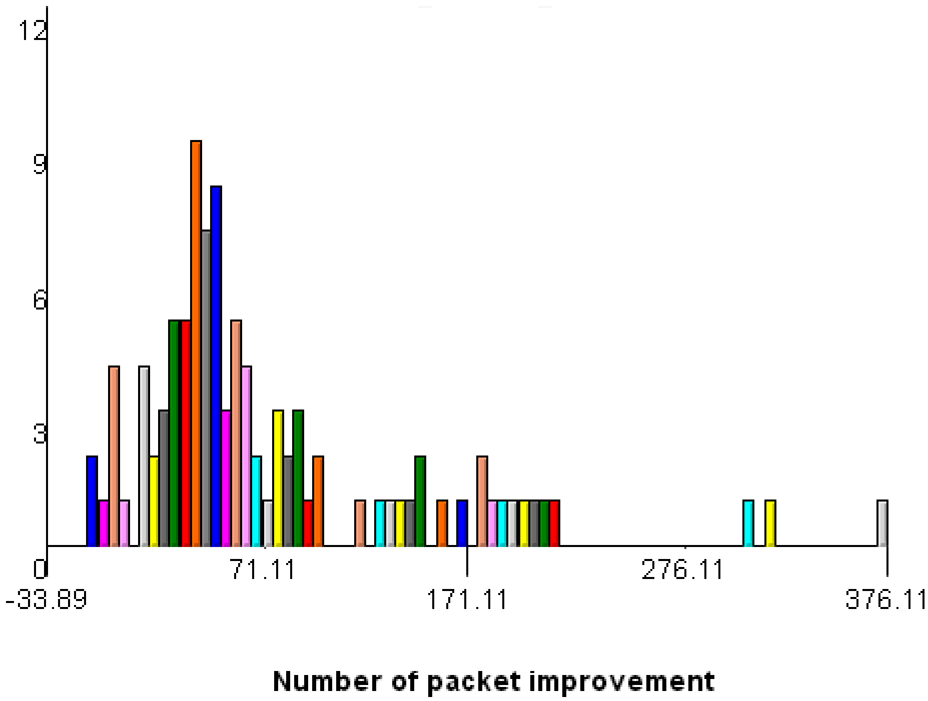


Figure II. Number of packet improvement percentage histogram of 100 randomly deployed networks with the second fuzzy logic control function.

Fuzzy logic control has a positive impact on WSN energy utilization

Similar to the packet transmission, 100 topologies were used to calculate the energy consumption statistically. The result is shown in Appendix Table II. E_0 is the initial energy of the whole network. E_{BS_NC}, E_{BS_WC1}, and E_{BS_WC2} are energy leftover and leftover percentage compared to E_0 either with no fuzzy function, with the first fuzzy function, or with the second fuzzy function, respectively. Energy leftover percentage histograms are made and shown in Figure III, IV and V. As can be seen from the figure, if no fuzzy control is used, the network become paralyzed when there is still about 80% energy left in the system. However, if fuzzy control is applied, because of the improvement in the network connectivity, the total energy of the system can be better utilized and drop the leftover energy to about 40%. This means the energy utilization is doubled under fuzzy control. Energy consideration is known to be the most important criteria in handling WSN, our result has indicated that fuzzy control can greatly improve the energy utilization and thus system performance, resulting in a much better WSN connectivity control manipulation.

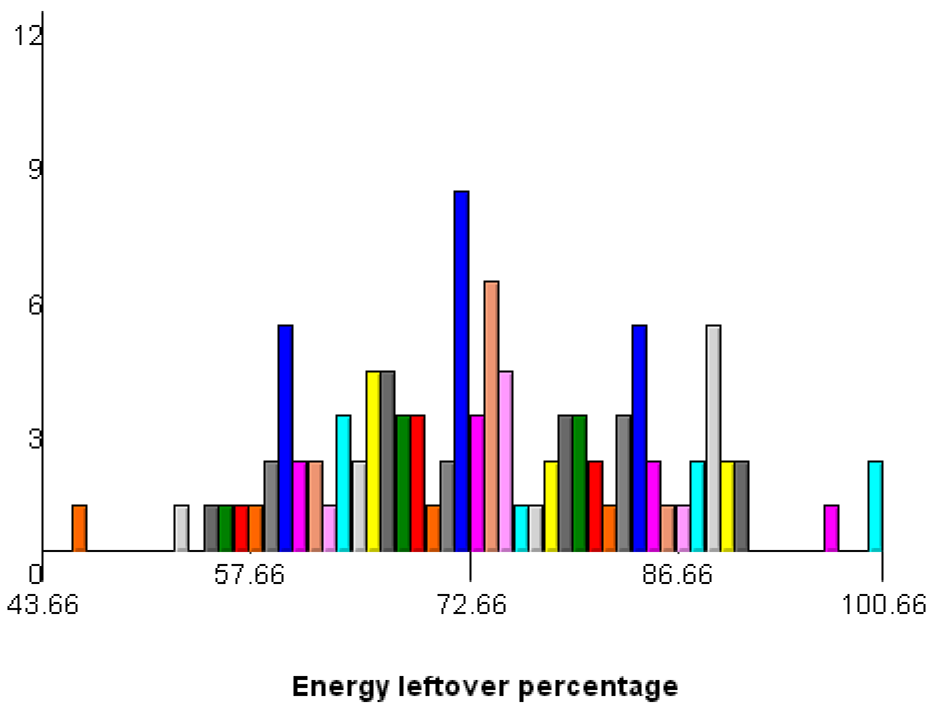


Figure III. Energy leftover percentage histogram of 100 randomly deployed networks without using fuzzy logic control.

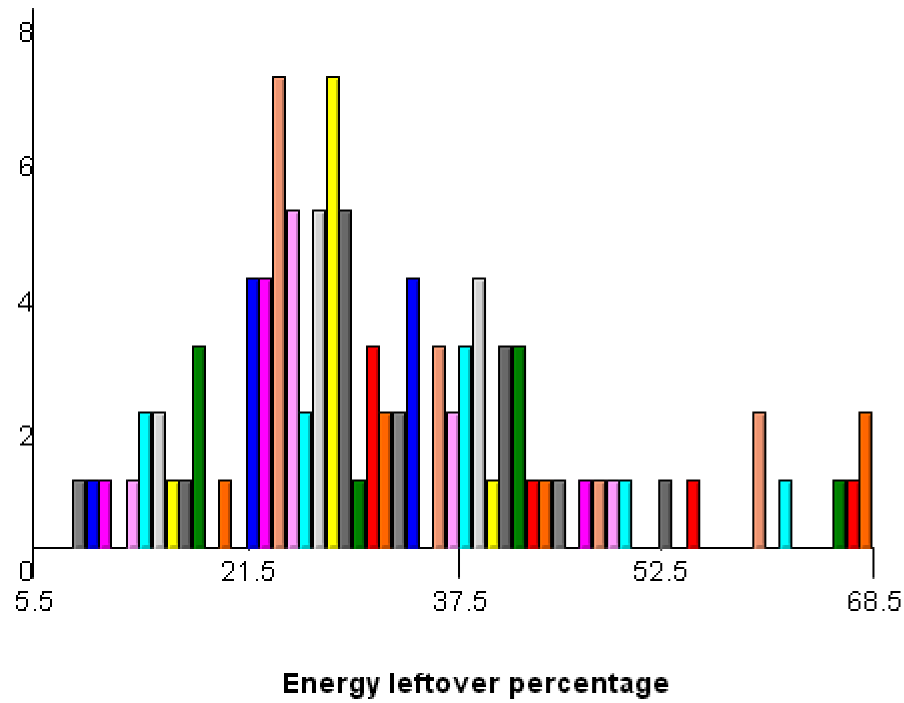


Figure IV. Energy leftover percentage histogram of 100 randomly deployed networks with the first fuzzy logic control function.

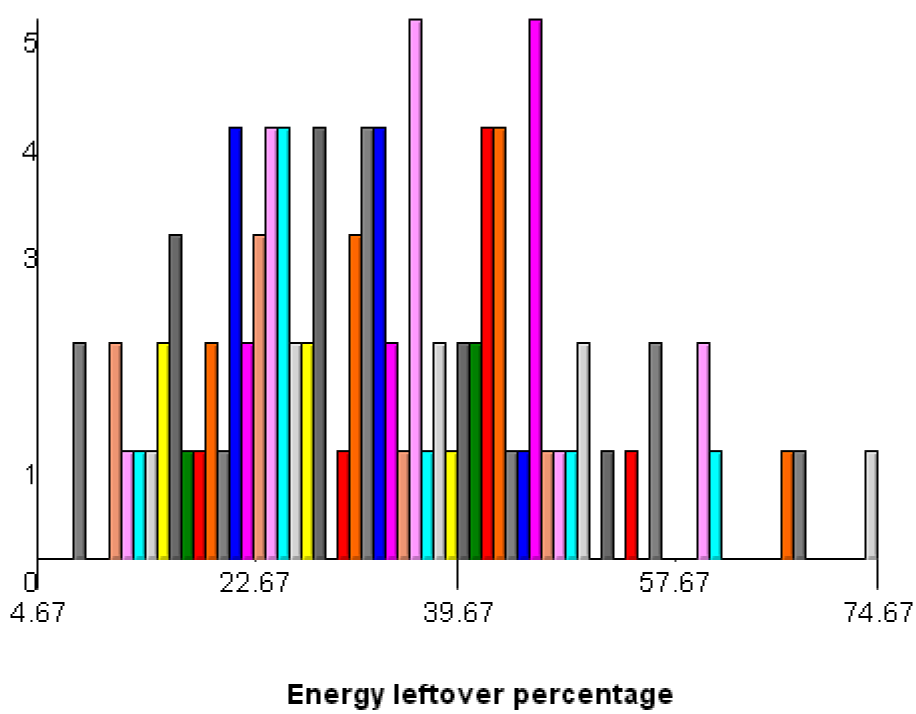


Figure V. Energy leftover percentage histogram of 100 randomly deployed networks with the second fuzzy logic control function.

Conclusion

The self-adaptive Wireless Sensor Network (WSN) based on Fuzzy Logic enables the nodes to automatically adjust the communication range in order to increase / decrease the node degrees (ND) of each node. This will directly improve the routing path of the packets to be delivered to the desired destination. In addition, the adjustment of the communication range is taken into consideration of the critical energy level of the individual node for the sake of energy saving. In our paper we have proved that the total number of packets received by the Base Station have been increased significantly comparing to the network without fuzzy logic. Also, the energy has been better utilized. In our experiment , we've proven that the topology of the node distribution is critical in the WSN, however by applying fuzzy logic, it won't become a bottleneck which is the case for the network without fuzzy logic. In conclusion, the network connectivity has been improved and therefore the network is more resilient to the nodes failure.

Recommendation for Future Studies

During our research and experiment, we learnt that there are still several questions that are still open. We've proven that the energy has been better utilized, but we didn't derive any result to prove the fuzzy logic can save more energy than without fuzzy logic. What's the optimal values for the fuzzy function parameters in order to achieve optimal balance between the performance of the network connectivity and the energy consumption. Also what's the optimal algorithms to follow for various network topologies? How to adjust the desired node degree for each node independently according to its location in the network? If we take into consideration of other environmental factors like weather, humidity that will influence the network, what's the effect to our fuzzy logic functions. We hope all of these questions will be answered in future works.

Bibliography

[1] A. A. Aziz,Y. A. Sekercioglu, P. Fitzpatrick, andM. Ivanovich, “A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks,” IEEE Communications Surveys and Tutorials, vol. 15,no. 1, pp. 121–144, 2013.

[2] C. Zhu, C. Zheng, L. Shu, and G. Han, “A survey on coverage and connectivity issues in wireless sensor networks,” Journal of Network and Computer Applications, vol. 35, no. 2, pp. 619–632, 2012.

[3] L. Atzori, A. Iera, and G. Morabito, “The internet of things: a survey,” Computer Networks, vol. 54,no. 15,pp. 2787–2805, 2010.

[4] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” IEEE Trans. Autom. Control, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[5] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” IEEE Trans. Autom. Control, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[6] W. Ren and R. Beard, “Consensus of information under dynamically changing interaction topologies,” in Proc. Amer. Control Conf., Boston, MA, Jun. 2004, pp. 4939–4944.

[7] J. Cortes, S. Martinez, and F. Bullo, “Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions,” IEEE Trans. Autom. Control, vol. 51, no. 8, pp. 1289–1298, Aug. 2006.

[8] J. Lin, A. S. Morse, and B. D. O. Anderson, “The multiagent rendezvous problem,” in Proc. 42nd IEEE Conf. Decision Control, Maui, HI, Dec. 2003, pp. 1508–1513.

[9] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, “Flocking in fixed and switching networks,” IEEE Trans. Autom. Control, vol. 52, no. 5, pp. 863–868, May 2007.

[10] R. Sepulchre, D. Paley, and N. E. Leonard, “Stabilization of planar collective motion: All-to-all communication,” IEEE Trans. Autom. Control, vol. 52, no. 5, pp. 811–824, May 2007.

[11] M. M. Zavlanos and G. J. Pappas, “Dynamic assignment in distributed motion planning with local coordination,” IEEE Trans. Robot., vol. 24, no. 1, pp. 232–242, Feb. 2008.

[12] J. P. Desai, J. P. Ostrowski, and V. Kumar, “Modeling and control of formations of nonholonomic mobile robots,” IEEE Trans. Robot. Autom., vol. 17, no. 6, pp. 905–908, Dec. 2001.

[13] G. Lafferriere, A. Williams, J. Caughman, and J. J. P. Veerman, “Decentralized control of vehicle formations,” Syst. Control Lett., vol. 54, no. 9, pp. 899–910, Sep. 2005.

[14] T. Balch and R. C. Arkin, “Behavior-based formation control for multirobot teams,” IEEE Trans. Robot. Autom., vol. 14, no. 6, pp. 926–939, Dec. 1998.

[15] P. Ogren, M. Egerstedt, andX.Hu, “A control Lyapunov function approach to multiagent coordination,” IEEE Trans. Robot. Autom., vol. 18, no. 5, pp. 847–851, Oct. 2002.

[16] K. E. Bekris,A.A.Argyros, and L. E. Kavraki, “New methods for reaching the entire plane with angle-based navigation,” in Proc. IEEE Int. Conf. Robot. Autom, New Orleans, LA, Apr. 2004, pp. 2373–2378.

[17] S. Poduri and G. S. Sukhatme, “Constrained coverage for mobile sensor networks,” in Proc. IEEE Int. Conf. Robot. Autom., New Orleans, LA, May 2004, pp. 165–172.

[18] M. Mesbahi, “On state-dependent dynamic graphs and their controllability properties,” IEEE Trans. Autom. Control, vol. 50, no. 3, pp. 387–392, Mar. 2005.

[19] D. P. Spanos and R. M. Murray, “Robust connectivity of networked vehicles,” in Proc. 43rd IEEE Conf. Decision Control, Bahamas, Dec. 2004, pp. 2893–2898.

[20] M. Ji and M. Egerstedt, “Distributed formation control while preserving connectedness,” in Proc. 45th IEEE Conf. Decision Control, San Diego, CA, Dec. 2006, pp. 5962–5967.

[21] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian,” IEEE Trans. Autom. Control, vol. 51, no. 1, pp. 116–120, Jan. 2006.

[22] M. C. DeGennaro andA. Jadbabaie, “Decentralized control of connectivity for multiagent systems,” in Proc. 45th IEEE Conf. Decision Control, San Diego, CA, Dec. 2006, pp. 3628–3633.

[23] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie, “Maintaining limited-range connectivity among second-order agents,” in Proc. Amer. Control Conf., Minneapolis, MN, Jun. 2006, pp. 2124–2129.

[24] L. Li, J. Y. Halpern, P. Bahl, Y. M. Wang, and R. Wattenhofer, “A conebased distributed topology-control algorithm for wireless multi-hop networks,” IEEE/ACM Trans. Netw., vol. 13, no. 1, pp. 147–159, Feb. 2005.

[25] R. D’Souza, D. Galvin, C. Moore, and D. Randall, “Global connectivity from local geometric constraints for sensor networks with various wireless footprints,” in Proc. 5th Int. Conf. Inf. Process. Sensor Netw., Nashville, TN, Apr. 2006, pp. 19–26.

[26] R. Wattenhofer and A. Zollinger, “XTC: A practical topology control algorithm for Ad-Hoc networks,” in Proc. 18th IEEE Int. Parallel Distrib. Process. Symp., Santa Fe, NM, Apr. 2004, pp. 216–223.

[27] M. M. Zavlanos, A. Tahbaz-Salehi, A. Jadbabaie, and G. J. Pappas, “Distributed topology control of dynamic networks,” in Proc. Amer. Control Conf., Seattle, WA, Jun. 2008, pp. 2660–2665.

[28] M. M. Zavlanos andG. J. Pappas, “Potential fields for maintaining connectivity of mobile networks,” IEEE Trans. Robot., vol. 23, no. 4, pp. 812–816, Aug. 2007.

[29] M. M. Zavlanos and G. J. Pappas, “Controlling connectivity of dynamic graphs,” in Proc. 44th IEEE Conf. Decision Control, Seville, Spain, Dec. 2005, pp. 6388–6393.

[30] J. Cortes, S. Martinez, and F. Bullo, “Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions,” IEEE Transactions on Automatic Control, vol. 51, no. 8, pp. 1289–1298, August 2006.

[31] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” IEEE Transactions on Automatic Control, vol. 48, no. 6, pp. 988–1001, June 2003.

[32] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, “Distributed topology control for wireless multihop ad-hoc networks,” in INFOCOM, 2001, pp. 1388–1397. [Online]. Available: citeseer.ist.psu.edu/wattenhofer01distributed.html

[33] A. Muhammad and M. Egerstedt, “Connectivity graphs as models of local interactions,” Proceedings of the 43th IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, December 2004.

[34] D. Spanos and R. M. Murray, “Robust connectivity of networked vehicles,” Proceedings of the 43th IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, December 2004.

[35] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph laplacian,” IEEE Transactions on Automatic Control, vol. 51 no. 1, pp. 116-120, January 2006.

[36] M. M. Zavlanos and G. J. Pappas, “Controlling connectivity of dynamic graphs,” Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, December 2005.

[37] Y. Huang, J.-F. Mart´ınez, J. Sendra, and L. L´opez, “The influence of communication range on connectivity for resilient wireless sensor networks using a probabilistic approach,” International Journal of Distributed Sensor Networks, vol. 2013, 2013.

[38] H. Bagci and A. Yazici, “An energy aware fuzzy approach to unequal clustering in wireless sensor networks,” Applied Soft Computing, vol. 13, no. 4, pp. 1741 – 1749, 2013.

Appendix

Trial	E_0	E_{BS_NC}	NC Leftover Percentage	E_{BS_WC1}	WC1 Leftover Percentage	E_{BS_WC2}	WC2 Leftover Percentage
1	30000	17912	59.71%	8036	26.79%	9833	32.78%
2	30000	27402	91.34%	19662	65.54%	22261	74.20%
3	30000	23049	76.83%	20053	66.84%	20296	67.65%
4	30000	25398	84.66%	11404	38.01%	11553	38.51%
5	30000	27069	90.23%	18487	61.62%	16915	56.38%
6	30000	22274	74.25%	12219	40.73%	12990	43.30%
7	30000	19738	65.79%	5072	16.91%	4572	15.24%
8	30000	22041	73.47%	10006	33.35%	7089	23.63%
9	30000	17337	57.79%	6933	23.11%	6363	21.21%
10	30000	16405	54.68%	6716	22.39%	8446	28.15%
11	30000	24096	80.32%	5382	17.94%	4725	15.75%
12	30000	21717	72.39%	10763	35.88%	15586	51.95%
13	30000	16916	56.39%	6464	21.55%	8160	27.20%
14	30000	26976	89.92%	10198	33.99%	11897	39.66%
15	30000	18932	63.11%	9198	30.66%	9649	32.16%
16	30000	17980	59.93%	6682	22.27%	6804	22.68%
17	30000	19328	64.43%	3375	11.25%	3235	10.78%
18	30000	17866	59.55%	4532	15.11%	4566	15.22%
19	30000	17730	59.10%	10120	33.73%	9290	30.97%
20	30000	21657	72.19%	8339	27.80%	9234	30.78%
21	30000	21750	72.50%	7252	24.17%	5171	17.24%
22	30000	25398	84.66%	8372	27.91%	12286	40.95%
23	30000	18678	62.26%	8333	27.78%	8369	27.90%
24	30000	19523	65.08%	3975	13.25%	6459	21.53%
25	30000	23381	77.94%	8445	28.15%	9836	32.79%
26	30000	21755	72.52%	11908	39.69%	9267	30.89%
27	30000	21942	73.14%	7727	25.76%	11296	37.65%
28	30000	26726	89.09%	6831	22.77%	6885	22.95%
29	30000	17067	56.89%	8736	29.12%	11334	37.78%
30	30000	23887	79.62%	8536	28.45%	7573	25.24%
31	30000	18156	60.52%	8254	27.51%	5623	18.74%
32	30000	29160	97.20%	14033	46.78%	14308	47.69%
33	30000	26514	88.38%	14337	47.79%	13073	43.58%

34	30000	24001	80.00%	8293	27.64%	8571	28.57%
35	30000	22618	75.39%	4692	15.64%	4040	13.47%
36	30000	25086	83.62%	11093	36.98%	10784	35.95%
37	30000	24358	81.19%	11252	37.51%	13768	45.89%
38	30000	21719	72.40%	7590	25.30%	7186	23.95%
39	30000	19784	65.95%	12211	40.70%	13262	44.21%
40	30000	22665	75.55%	9982	33.27%	10283	34.28%
41	30000	20387	67.96%	8766	29.22%	6725	22.42%
42	30000	22236	74.12%	20311	67.70%	18233	60.78%
43	30000	26780	89.27%	9408	31.36%	12979	43.26%
44	30000	20159	67.20%	14925	49.75%	18019	60.06%
45	30000	15993	53.31%	6894	22.98%	3463	11.54%
46	30000	20272	67.57%	5386	17.95%	4766	15.89%
47	30000	23810	79.37%	11293	37.64%	12753	42.51%
48	30000	20826	69.42%	7546	25.15%	8138	27.13%
49	30000	21694	72.31%	8142	27.14%	5791	19.30%
50	30000	21665	72.22%	8598	28.66%	7561	25.20%
51	30000	20412	68.04%	7408	24.69%	7913	26.38%
52	30000	26608	88.69%	17940	59.80%	13894	46.31%
53	30000	19398	64.66%	8145	27.15%	7569	25.23%
54	30000	22917	76.39%	7101	23.67%	16988	56.63%
55	30000	23861	79.54%	10286	34.29%	9945	33.15%
56	30000	22611	75.37%	5939	19.80%	6444	21.48%
57	30000	30000	100.00%	8024	26.75%	9503	31.68%
58	30000	26832	89.44%	7944	26.48%	6499	21.66%
59	30000	18419	61.40%	5265	17.55%	4881	16.27%
60	30000	21895	72.98%	9589	31.96%	9895	32.98%
61	30000	20639	68.80%	7313	24.38%	9176	30.59%
62	30000	21199	70.66%	4099	13.66%	5438	18.13%
63	30000	25246	84.15%	14565	48.55%	15044	50.15%
64	30000	21345	71.15%	6798	22.66%	7123	23.74%
65	30000	26360	87.87%	12756	42.52%	12418	41.39%
66	30000	25127	83.76%	7322	24.41%	7945	26.48%
67	30000	25516	85.05%	12592	41.97%	13744	45.81%
68	30000	22195	73.98%	8978	29.93%	10223	34.08%
69	30000	13988	46.63%	2954	9.85%	2561	8.54%
70	30000	19108	63.69%	6608	22.03%	6541	21.80%
71	30000	30000	100.00%	20343	67.81%	20384	67.95%

72	30000	25530	85.10%	7316	24.39%	9532	31.77%
73	30000	18663	62.21%	4247	14.16%	2498	8.33%
74	30000	21774	72.58%	7489	24.96%	7316	24.39%
75	30000	24960	83.20%	11665	38.88%	10427	34.76%
76	30000	24551	81.84%	15770	52.57%	13924	46.41%
77	30000	19673	65.58%	7991	26.64%	7621	25.40%
78	30000	19984	66.61%	17960	59.87%	18172	60.57%
79	30000	22362	74.54%	8711	29.04%	10905	36.35%
80	30000	18174	60.58%	10205	34.02%	10702	35.67%
81	30000	27435	91.45%	8601	28.67%	14130	47.10%
82	30000	23562	78.54%	12285	40.95%	11983	39.94%
83	30000	26001	86.67%	12742	42.47%	16316	54.39%
84	30000	22357	74.52%	13057	43.52%	12589	41.96%
85	30000	24237	80.79%	9640	32.13%	7319	24.40%
86	30000	26622	88.74%	9262	30.87%	12646	42.15%
87	30000	21145	70.48%	13549	45.16%	13765	45.88%
88	30000	24889	82.96%	11566	38.55%	13607	45.36%
89	30000	25947	86.49%	16470	54.90%	14853	49.51%
90	30000	22297	74.32%	7357	24.52%	9508	31.69%
91	30000	20130	67.10%	4641	15.47%	4151	13.84%
92	30000	20438	68.13%	7199	24.00%	8354	27.85%
93	30000	18162	60.54%	2634	8.78%	3551	11.84%
94	30000	18322	61.07%	10931	36.44%	12971	43.24%
95	30000	20778	69.26%	7202	24.01%	6089	20.30%
96	30000	25270	84.23%	11733	39.11%	12030	40.10%
97	30000	23960	79.87%	10908	36.36%	10886	36.29%
98	30000	20168	67.23%	10980	36.60%	12764	42.55%
99	30000	22662	75.54%	12699	42.33%	15152	50.51%
100	30000	19755	65.85%	11578	38.59%	10999	36.66%
mean		22232.97	74.11%	9651.19	32.17%	10101.54	33.67%
STD		3292.854	10.98%	3909.051	13.03%	4225.339	14.08%

Table I. Comparison of received packets at base station (average of 100 networks randomly deployed)

Trial	P_{BS_NC}	P_{BS_WC1}	WC1 Improvement	P_{BS_WC2}	WC2 Improvement
1	214	265	23.83%	246	14.95%
2	63	114	80.95%	95	50.79%
3	96	94	-2.08%	94	-2.08%
4	72	205	184.72%	206	186.11%
5	117	151	29.06%	161	37.61%
6	162	217	33.95%	224	38.27%
7	203	353	73.89%	354	74.38%
8	141	216	53.19%	229	62.41%
9	181	286	58.01%	268	48.07%
10	279	314	12.54%	330	18.28%
11	149	374	151.01%	374	151.01%
12	157	243	54.78%	203	29.30%
13	143	279	95.10%	253	76.92%
14	66	206	212.12%	205	210.61%
15	218	269	23.39%	262	20.18%
16	262	364	38.93%	361	37.79%
17	352	430	22.16%	406	15.34%
18	217	317	46.08%	317	46.08%
19	173	279	61.27%	282	63.01%
20	172	308	79.07%	312	81.40%
21	198	290	46.46%	309	56.06%
22	75	256	241.33%	224	198.67%
23	227	320	40.97%	309	36.12%
24	207	314	51.69%	304	46.86%
25	99	293	195.96%	279	181.82%
26	153	208	35.95%	231	50.98%
27	84	256	204.76%	233	177.38%
28	62	155	150.00%	155	150.00%
29	275	256	-6.91%	243	-11.64%
30	233	282	21.03%	287	23.18%
31	202	293	45.05%	315	55.94%
32	60	158	163.33%	162	170.00%
33	71	199	180.28%	217	205.63%
34	99	220	122.22%	214	116.16%
35	183	318	73.77%	304	66.12%

36	102	244	139.22%	246	141.18%
37	62	196	216.13%	193	211.29%
38	181	336	85.64%	345	90.61%
39	155	200	29.03%	193	24.52%
40	150	217	44.67%	230	53.33%
41	192	231	20.31%	249	29.69%
42	120	166	38.33%	203	69.17%
43	70	220	214.29%	205	192.86%
44	109	129	18.35%	122	11.93%
45	238	332	39.50%	341	43.28%
46	260	346	33.08%	371	42.69%
47	158	221	39.87%	219	38.61%
48	206	312	51.46%	327	58.74%
49	113	213	88.50%	212	87.61%
50	127	213	67.72%	224	76.38%
51	212	306	44.34%	316	49.06%
52	46	135	193.48%	128	178.26%
53	220	260	18.18%	287	30.45%
54	115	173	50.43%	113	-1.74%
55	149	232	55.70%	246	65.10%
56	186	308	65.59%	347	86.56%
57	0	267	Infinity	262	Infinity
58	132	250	89.39%	262	98.48%
59	186	304	63.44%	308	65.59%
60	113	235	107.96%	225	99.12%
61	198	283	42.93%	274	38.38%
62	286	359	25.52%	376	31.47%
63	116	174	50.00%	175	50.86%
64	175	272	55.43%	259	48.00%
65	88	163	85.23%	169	92.05%
66	80	323	303.75%	336	320.00%
67	118	168	42.37%	166	40.68%
68	147	189	28.57%	169	14.97%
69	348	497	42.82%	502	44.25%
70	219	230	5.02%	208	-5.02%
71	0	101	Infinity	101	Infinity
72	58	256	341.38%	238	310.34%
73	302	363	20.20%	405	34.11%

74	131	213	62.60%	233	77.86%
75	150	209	39.33%	220	46.67%
76	160	199	24.38%	232	45.00%
77	172	306	77.91%	318	84.88%
78	142	142	0.00%	141	-0.70%
79	130	155	19.23%	132	1.54%
80	200	279	39.50%	253	26.50%
81	90	179	98.89%	144	60.00%
82	127	117	-7.87%	126	-0.79%
83	57	172	201.75%	137	140.35%
84	167	214	28.14%	219	31.14%
85	121	284	134.71%	280	131.40%
86	144	224	55.56%	208	44.44%
87	249	336	34.94%	325	30.52%
88	134	217	61.94%	210	56.72%
89	39	169	333.33%	185	374.36%
90	196	251	28.06%	243	23.98%
91	205	305	48.78%	329	60.49%
92	302	390	29.14%	408	35.10%
93	299	486	62.54%	481	60.87%
94	258	236	-8.53%	226	-12.40%
95	175	254	45.14%	250	42.86%
96	100	206	106.00%	231	131.00%
97	85	218	156.47%	218	156.47%
98	148	206	39.19%	206	39.19%
99	155	228	47.10%	211	36.13%
100	214	306	42.99%	291	35.98%
mean	157.52	250.27	77.11%	249.77	75.57%
STD	72.53901	76.65268	73.08%	80.62143	72.12%

Table II. Comparison of energy leftover of the whole network (average of 100 networks randomly deployed)