Some Studies on Topology Management in Wireless Sensor Networks

Thesis submitted by

Chiranjib Patra

Doctor of Philosophy (Engineering)

Department of Information Technology
Faculty Council of Engineering and Technology
Jadavpur University
Kolkata, India

2018

JADAVPUR UNIVERSITY

KOLKATA – 700032, INDIA

INDEX NO.: 186/11/Engg.

Title of the Thesis:

Some Studies on Topology Management in Wireless Sensor Networks

Thesis submitted by: Chiranjib Patra

Name, Designation and Institution of the Supervisors/s

Supervisors:

1. Dr. Matangini Chattopadhyay

   Professor, School of Education Technology, Jadavpur University

2. Dr. Parama Bhaumik

   Associate Professor, Department of Information Technology, Jadavpur University

List of Publications

Conferences:

1. Patra, Chiranjib, Matangini Chattopadhyay, Parama Bhaumik, and Anjan Guha Roy. "Using self organizing map in wireless sensor network for designing energy efficient topologies." In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pp. 1-6. IEEE, 2011.

2. Patra, Chiranjib, Arindam Mondal, Parama Bhaumik, and Matangini Chattopadhyay. "An energy efficient event based hierarchical clustering algorithm for wireless sensor networks." *Global Trends in Computing and Communication Systems*(2012): 380-385.

3. Patra, Chiranjib, Matangini Chattopadhyay, Parama Bhaumik, Munmun Bhattacharya, and Saswati Mukherjee. "A reliable two-tier energy-efficient topology building algorithm for Wireless Sensor Networks." In *Applications and Innovations in Mobile Computing (AIMoC), 2014*, pp. 146-150. IEEE,2014.

4. Patra, Chiranjib, and Botezatu, Nocolea "Effect of gossiping on some basic wireless sensor network protocols",2017 21st International Conference on System Theory, Control and Computing (ICSTCC) 2017.

5. Patra, Chiranjib and Botezatu, Niclolea, "Improving the Performance of Hierarchical Clustering Protocols with Network Evolution Model", Springer Series: Advances in Intelligent Systems and Computing - ISSN 2194-5357,2018

Journals:

1. Patra, Chiranjib, Anjan Guha Roy, Samiran Chattopadhyay, and Parama Bhaumik. "Designing energyefficient topologies for wireless sensor network: neural approach." *International Journal of Distributed Sensor Networks* 6, no. 1 (2010): 216716.
2. Patra, Chiranjib, Samiran Chattopadhyay, Matangini Chattopadhyay, and Parama Bhaumik. "Analysing topology control protocols in wireless sensor network using network evolution model." *International Journal of Distributed Sensor Networks* 11, no. 10 (2015): 693602.

Book Chapters:

1. Patra, Chiranjib, Arindam Mondal, Parama Bhaumik, and Matangini Chattopadhyay. "Topology Management in Wireless Sensor Networks." In *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management*, pp. 14-24. IGI Global, 2012.

Conferences and workshops attended:

1. *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011*
2. *Applications and Innovations in Mobile Computing (AIMoC) 2014*
3. 21st International Conference on System Theory, Control and Computing (ICSTCC) 2017

Certificate from the Supervisors

This is to certify that the thesis entitled "Some Studies on Topology Management in Wireless Sensor Networks" submitted by Shri Chiranjib Patra, who got his name registered on 10th June 2011 for the PhD (Engineering) degree of Jadavpur university is absolutely based upon his own work under the supervision of Dr. Matangini Chattopadhyay, Professor, School of Education Technology, Jadavpur University and Dr. Parama Bhaumik Associate Professor, Department of Information Technology, Jadavpur University. And that his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

1…………………………………………..

Signature of the Supervisor and date with Office Seal

2…………………………………………..

Signature of the Supervisor and date with Office Seal

Acknowledgements

The last seven years have been part of an experience in which I have gratefully had the opportunity to grow not only as a professional but as a human being.

I will start by thanking Prof. Samiran Chattopadhyay and advisors Prof. Matangini Chattopadhyay and Prof. Parama Bhaumik, who has guided and supported me throughout these years with admirable patience.

Thanks to my Erasmus supervisiors Dr. Nikolea Botezatu and Prof. Vaslie Ion Manta for accepting my invitation to be part of this experience and for their valuable comments, insights and suggestions.

My deepest gratitude to my wife Mahuya and my beloved daughter Upasana, whose love, patience and support have helped to keep me on the right track. I extend this gratitude especially to my mother for her unconditional love, and to my parents-in-law without whose help I would not have had the opportunity to finish this dissertation.

Thanks to all my friends who I no longer consider just friends, because we have become more like a family: thanks to Debdutta, Sourish, Samir and all the people from Department of Information Technology , Calcutta Institute of Engineering and Management, India for their logistic and mental support on this endeavor. I would like to extend my thanks to Prof. B.C Dhara for providing excellent suggestions during final months of preparing the thesis.

Last not the least, thanks also to all the staff especially Mr. Susanta Chattopadhyay at the Department of Information Technology at Jadavpur University, India and Department of Computer Science , Gheoghe Asachi Technical University, Romania who have been always there for me.

Finally, and most importantly, I thank God for giving me the opportunity to grow and improve myself in order to be able to serve better.

Chiranjib Patra
Salt Lake, Kolkata

Abstract

Wireless Sensor Networks (WSN) offer a flexible low-cost solution to the problem of event monitoring, especially in places with limited accessibility or that represent danger to humans. WSNs are made of resource-constrained wireless devices, which require energy efficient mechanisms, algorithms and protocols. One of these mechanisms is Topology Management (TM).The TM mechanism can be executed in three ways Topology Discovery, Clustering and Duty Cycle. In order to sustain the topology, perseverance is one of the mechanism that enables the life time and better delivery of the packets in spite of common inevitable bottlenecks like congestion.

This dissertation expands the knowledge of TM mainly in three sections as mentioned above. Firstly in the topology discovery section, it introduces the concept of partitioning the number of sensors in the test area in such a way that the connectivity and coverage of the network is preserved. The TM in this algorithm is implemented using Self Organizing maps along with nearest neighbor algorithm .Secondly in the clustering section, it introduces to the concept of common minimum transmission level and knowledge based inference approach to select cluster heads as the part of TM using clustering. This section also introduces a model for large scale sensor networks called network evolution model using the concept of scale free networks. This network evolution model is used to evaluate the probability of clustering in wireless sensor network (WSN).This probability framework thus obtained considerably agrees between the experimental and theoretical values. Again using the concept of network evolution model, it is found to improve the life time of the network to 150 to 60 percentage in case of LEACH and LEACH-C protocols. It is also observed that as the number of nodes increases, the better topology management is done and hence the better life time of the WSN. In preserving the topology as created by topology discovery and clustering, perseverance of the topology is done by reducing the message complexity. This optimization process is done by implementing the gossiping framework which showed acceptable promise of reducing the message overhead by 25 percent in case of some basic routing protocols like span tree and angular routing.

List of Abbreviations

ABCP	Access-Based Clustering Protocol
ABEE	Access-Based Energy Efficient
ADB	Asynchronous Duty-cycle Broadcasting
ANN	Approximate Nearest Neighbors
APTEEN	A Hybrid Protocol for Efficient Routing
BCDCP	Base-Station Controlled Dynamic Clustering Protocol
CBTC	Cone-Based Topology Control
CDS	Connected Dominating Set
CMM	Clustered Mobility Model
DRNG	Directed Relative Neighborhood Graph
EECDS	Energy Efficient Connected Dominating
EECS	Energy Efficient Clustering Scheme
GLIDER	Gradient Landmark-based Distributed Routing
HMS	Heuristic Multidimensional Scaling
KNN	K-Nearest Neighbours
LEACH	Low-Energy Adaptive Clustering Hierarchy
MANET	Mobile Ad-hoc NETworks
MST	Minimum Spanning Tree
NLT	Network Life Time
OFLDCUL	Opportunistic Flooding in Low Duty Cycle with Unreliable Links
PEGASIS	Power-Efficient. GAthering in Sensor Information Systems
PROWLER	Probabilistic Wireless Network Simulator
RTTEE	Reliable Two Tier Energy Efficient
SOFM	Self Organising Maps
SOM	Self Organizing Map
STREAM	Sensor Topology Retrieval at Multiple Resolutions
UDG	Unit Disk Graph
WSN	Wireless Sensor Network

List of Tables

List of Figures

Tables of Contents

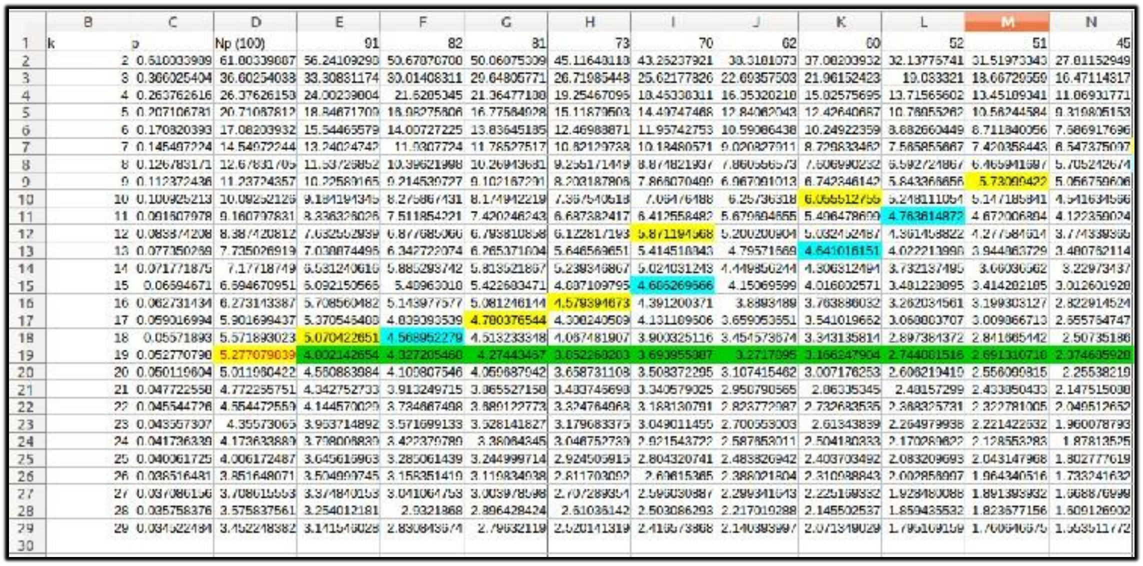
APPENDIX A

The optimization matrix is a spreadsheet-based numerical analysis for the changing of the alive nodes in the experiment. We will discuss the matrix used in the figure. Columns B and C represent the number of connections per cluster and the corresponding probability as calculated by equation 1. The rest of the columns are the multiplication of the corresponding probability (due to links) to the number of nodes. In this way, the cells are populated, and the matrix is created.

To plot the cluster heads, for example in the case of the variable clustering LEACH-C, we get the following results from Table 6 which is plotted in the optimization matrix as yellow. Similarly, we plot the original LEACH-C with the following results from Table 6, and the plot is shown in cyan.

We can see that the plot follows diagonal lines, which implies that the system is taking the longest performance path. Due to rapidly changing cluster heads, the setup phase of LEACH-C does not perform if the schema of changing cluster heads is done above the yellow line, as seen in the optimizing matrix.

If we attempt to move below the green line, it is against the system under consideration because the LEACH-C protocol operates with a decreasing number of alive nodes. It can also be seen that the sum of all the cluster heads produced along the green line is lower compared to the previous amounts obtained in the preceding rows. Therefore, in the number of (alive node - cluster) spaces, this green line is the optimized cluster head varying scheme.



	B	C	D	E	F	G	H	I	J	K	L	M	N
1	k	p	Np (100)	91	82	81	73	70	62	60	52	51	45
2	2	0.610033909	61.00339087	56.24109290	50.67070700	50.06075309	45.11640110	43.25237921	38.3181073	37.08200932	32.13775741	31.51973343	27.81152949
3	3	0.366025404	36.60254038	33.30831174	30.01408311	29.64805771	26.71985448	25.62177826	22.69357503	21.96152423	19.033321	18.66729559	16.47114317
4	4	0.263762616	26.37626153	24.00239004	21.6285345	21.36477108	19.25467095	18.46338311	16.35328218	15.82575695	13.71565602	13.45189341	11.86931771
5	5	0.207106781	20.71067812	18.84671709	16.98275806	16.77564928	15.11879503	14.49747468	12.84062043	12.42640687	10.76955262	10.56244584	9.319805153
6	6	0.170820393	17.08203932	15.54465579	14.00727225	13.83645185	12.46988871	11.95742753	10.59086438	10.24922359	8.882660449	8.711840056	7.686917696
7	7	0.145497224	14.54972244	13.24024742	11.9307724	11.78527517	10.62129738	10.18480571	9.020827911	8.729833462	7.565855667	7.420358443	6.547375097
8	8	0.126783171	12.67831705	11.53726852	10.39621998	10.26943681	9.255171449	8.874821937	7.860556573	7.606990232	6.592724867	6.465941697	5.705242674
9	9	0.112372436	11.23724357	10.22589165	9.214539727	9.102167291	8.203187806	7.866070499	6.967091013	6.742346142	5.843366656	5.73099422	5.056759606
10	10	0.100925213	10.09252126	9.184194345	8.275867431	8.174942219	7.367540518	7.06476488	6.25736318	6.055512755	5.248111054	5.147185841	4.541634566
11	11	0.091607978	9.160797831	8.336326026	7.511854221	7.420246243	6.687382417	6.412558482	5.679694655	5.496478699	4.763614872	4.672006894	4.122359024
12	12	0.083874208	8.387420812	7.632552939	6.877685066	6.793810858	6.122817193	5.871194568	5.200200904	5.032452487	4.361458822	4.277584614	3.774339365
13	13	0.077350269	7.735026919	7.038874496	6.342722074	6.265371804	5.646569651	5.414518843	4.79571669	4.641016151	4.022213998	3.944863729	3.480762114
14	14	0.071771875	7.17718749	6.531240616	5.885293742	5.813521867	5.239346867	5.024031243	4.449856244	4.306312494	3.732137495	3.66036562	3.22973437
15	15	0.06694671	6.694670951	6.092150565	5.48963018	5.422683471	4.887109795	4.686269666	4.15069599	4.016802571	3.481228895	3.414282185	3.012601928
16	16	0.062731434	6.273143387	5.708560482	5.143977577	5.081246144	4.579394673	4.391200371	3.8893489	3.763886032	3.262034561	3.199303127	2.822914524
17	17	0.059016994	5.901699437	5.370546488	4.839393539	4.780376544	4.308240589	4.131189606	3.659053651	3.541019662	3.068883707	3.009866713	2.655764747
18	18	0.05571893	5.571893023	5.070422651	4.568952279	4.513233348	4.067481907	3.900325116	3.454573674	3.343135814	2.897384372	2.841665442	2.50735186
19	19	0.052770790	5.277079839	4.802142654	4.327205460	4.27443467	3.852268283	3.693955887	3.2717895	3.166247904	2.744081516	2.691310718	2.374685928
20	20	0.050119604	5.011960422	4.560883984	4.109807546	4.059687942	3.658731108	3.508372295	3.107415462	3.007176253	2.606219419	2.556099815	2.25538219
21	21	0.047722558	4.772255751	4.342752733	3.913249715	3.865527158	3.483746698	3.340579025	2.958798565	2.86335345	2.48157299	2.433850433	2.147515088
22	22	0.045544726	4.554472559	4.144570029	3.734667498	3.689122773	3.324764968	3.188130791	2.823772987	2.732683535	2.368325731	2.322781005	2.049512652
23	23	0.043557307	4.35573065	3.963714892	3.571699133	3.528141827	3.179683375	3.049011455	2.700553003	2.61343839	2.264979938	2.221422532	1.960078793
24	24	0.041736339	4.173633889	3.798006839	3.422379789	3.38064345	3.046752739	2.921543722	2.587653011	2.504180333	2.170289622	2.128553283	1.87813525
25	25	0.040061725	4.006172487	3.645616963	3.285061439	3.244999714	2.924505915	2.804320741	2.483826942	2.403703492	2.083209693	2.043147968	1.802777619
26	26	0.038516481	3.851648071	3.504999745	3.158351419	3.119834938	2.811703092	2.69615365	2.388021804	2.310988843	2.002856997	1.964340516	1.733241632
27	27	0.037086156	3.708615553	3.374840153	3.041064753	3.003978598	2.707289354	2.596030887	2.299341643	2.225169332	1.928480088	1.891393932	1.668876999
28	28	0.035758376	3.575837561	3.254012181	2.9321868	2.896428424	2.61036142	2.503086293	2.217019288	2.145502537	1.859435532	1.823677156	1.609126902
29	29	0.034522484	3.452248382	3.141546028	2.830843674	2.79632119	2.520141319	2.416573868	2.140393997	2.071349029	1.795169159	1.760646675	1.553511772
30													

Figure 6.8: The optimizing matrix

APPENDIX B

Systematic cluster head variation Scheme-I : As the total number of clusters obtained is N_C = p*, the total number of alive nodes (Na) , as the number of nodes (alive nodes) varies wrt time, so we have the total nodes as alive nodes, subsequently p (the probability of clustering) can be obtained with the value of k=((total number of alive nodes(Na))/(desired number of CH(n))) -1.

Mathematically this can be written as

$$Nc = \frac{1}{2}\left[\sqrt{\frac{k+3}{k-1}} - 1\right]*Na$$

$$Nc = \frac{1}{2}[\surd((\frac{Na}{n} - 1) + 3)/(((\frac{Na}{n} - 1)) - 1)) - 1]*Na$$

The relationship is clearly not linear. Therefore it attributes to a long path movement in the Nc vs. Na workspace (yellow in figure 7).

Systematic cluster head variation Scheme-II : As the total number of clusters obtained is as N_C = p*, the total number of alive nodes (Na), as the number of nodes (alive nodes) varies wrt time, so we have the total nodes as alive nodes, subsequently p (the probability of clustering) can be obtained with the value of k= ((total number of nodes(N))/(desired number of CH)) -1

Similarly, in this case, we have,

$$Nc = \frac{1}{2}[\surd((\frac{N}{n} - 1) + 3)/(((\frac{N}{n} - 1)) - 1)) - 1]*Na$$

The above equation shows the linear relation between Nc vs. Na . Therefore, the shortest path movement can be observed in the Nc vs. Na workspace (blue in figure 6.7).

Out of the above two schemes, scheme –II provides a better optimization and use of the sensor node energy during the LEACH clustering, which can be visualized in Figure 6.7 .

APPENDIX C

Assume the number of clusters (100 node network)	Number of connection per cluster	Clustering probability P=0.5[((k+3)/(k-1))^0.5 -1]	Number of theoretical clusters=p*100	Number of cluster round off
1	(100-1)=99	0.010102	1.01	1
2	100/2 – 1 =49	0.0204165	2.04	2
3	100/3 -1 = 32.33 = 32	0.031279	3.1	3
4	100/4 -1 = 24	0.041736	4.2	4
5	100/5 -1 = 19	0.0527	5.2	5
6	100/6 -1 = 15.66 = 16	0.06273	6.2	6
7	100/7 -1 =13.28=13	0.0773502	7.7	8

Table-C1 *Theoretical calculation of the number of cluster heads of a 100-node network assuming in the calculation as for example (4.2 as 4 and 4.6 as 5 ie set value wrt +0.5). Clearly, the seventh row violates the number between the first column and the fifth column.*

Similarly, for a 200-node network we have,

Assume the number of clusters (100 node network)	Number of connection per cluster	Clustering probability P=0.5[((k +3)/(k-1))^0.5 -1]	Number of theoretical clusters=p*100	Number of cluster round off
1	(200-1)=199	0.00502	1.005	1
2	200/2 – 1 =99	0.010102	2.02	2
3	200/3 -1=65.66= 66	0.015154 9	3.03	3
4	200/4 -1=49	0.020416 5	4.08	4
5	200/5 -1=39	0.02565	5.131496	5

6	200/6 -1=32.33=32	0.03127965	6.255	6
7	200/7 -1 =27.57=28	0.03575838	7.15	7
8	200/8 -1 =21.22=21	004173634	8.34	8
9	200/9 -1	0.04772256	9.5	10

Table-C2 *Theoretical calculation of the number of cluster heads of a 200-node network assuming in the calculation as for example (4.2 as 4 and 4.6 as 5 ie set value wrt +0.5). Clearly the seventh row violates the number between the first column and the fifth column.*

APPENDIX D

1.The Radio Propagation Model (RPM): The RPM determines the strength of a transmitted signal at a particular point of the space for all transmitters in the system. This strength is the measure of how far the node can communicate and how much energy will be spent by the system.

For a deterministic mode of transmission, the strength of the signal is defined as:

$P_d = P_{transmit}\ 1/(1+d^y)$

Where P_d is the ideal reception strength, $P_{transmit}$ is the transmission strength, d is the distance between the transmitter and the receiver and y is the decay coefficient. The range of y lies between 2 and 4.

However, real signals do have a fading effect and some un-modelled disturbances (external interference, hardware problems, etc.)

So the real strength will be $P_{error}+ P_{d,}$ where P_{error} is the strength loss due to un-modelled effects.

On the basis of RPM, the signal reception and collision are determined.

Model 1:

If both the receiver and the transmitter signals overlap, then collisions occur. If no signal is received, then the channel is sensed to be idle, and the signal can be received if it’s greater than the threshold reception parameter.

Model 2: If the SINR at the receiver is larger than the receiving limit during the whole period of transmission, the signal is said to have received. If the signal strength is smaller than the idle limit, then the channel is said to be idle. If the SINR at the receiver is smaller than the reception limit, the channel is said to have experienced collision.

It is found during the implementation that Model 1 is fast but Model 2 is accurate.

MAC layer:

On the lines of RPM, the MAC layer checks for the Channel idle, Channel transmission and Channel collision. This checking is done by sending the end packet event for random waiting time; if there is no response the channel is idle. Before each idle check waiting for random intervals, there is a characteristic called backoff time. After the reception of a packet on the receiver's side, the application receives a packet received or collided packet received event, depending on the success of the transmission. The waiting time and backoff time parameters are random uniformly distributed variables in predefined intervals, while transmission time is constant.

Application Layer:

The following functions/ methods are implemented to detail the nuances of the application layer InitApplication, PacketReceived ,PacketSent, CollidedPacket, ClockTick. These functions follow the concepts of RPM for the application layer.

4. Optimization layer:

Let us consider some points in P-Space as given by P={Pi}, i = 1..N, where N is the number of points. Considering all the points the average cost function Z={Zi} is maintained. The current and the previous points are denoted by Pc and Pp, respectively. If D is the vector for the set of values for the vectors such as left, right, up and down, by combining the P-Space with the vector D, one can have a step function such as $S=\{P,D\}$, which describes the actual topology. The function call is denoted by $F: P \rightarrow R$

In order to find the optimum topology, we use the following algorithm:

1. Initialization:

Zi=0, for all i. Pc = Pp = P0 (initialization).

2. Search:

F(Z) = F(Pc). Update Zc using FZ.

3. Step:

if FZ < Zp then keep the vector D (except for the outermost points),

Else

Choose the vector D on a random basis.

End

Pp = Pc, Pc = S(D, Pc).

4. Repeat 2 and 3 until the exit criterion is met.

The exit criterion is met if the number of iterations is fixed or manually changed for different cases.

The above algorithm implements a search method that uses a noisy cost function value, which returns different points when calculated over the same points. These calculated points actually represent the error surface more accurately.

Consider a situation of 100 nodes spread over the area A. One of the nodes initiates transmission. In response other nodes, those who receive, retransmit with probability p. If the transmit signal strength is s, then the goal of this simulation is to optimize p and s. So, the energy spent by the system is $E=A1*(100 - nt)^2 + A2*nr *s$. Now, running the optimizing algorithm for the search method, we obtain the values of p within the range of 0.1 to 1 and that of s as 0.1 to 5. The hops were entered manually for those nodes which lie in the straight path between the source and destination.

Bibliography

[1] A. J. Anto, J. Vasudevan, and P. Narayanasamy, "Multipath power sensitive routing protocol for mobile ad hoc networks," in Proceedings of the Conference Wireless on Demand Network Systems, vol. 2928 of Lecture Notes in Computer Science (LNCS), pp. 171–183, 2004.

[2] A.-L. Baraba´si and R. Albert, "Emergence of scaling in random networks", American Association for the Advancement of Science, vol.286, no. 5439, pp. 509–512, 1999 .H4,L7,

[3] Agarwal, D. P., & Manjeshwar, A. (2001). TEEN: a protocol for enhanced efficiency in wireless sensor networks. IEEE, San Francisco. 9. Manjeshwar, A., & Agrawal, D. P. (2002, April). APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In ipdps (p. 0195b). IEEE.

[4] Agarwal, P. K., Edelsbrunner, H., Schwarzkopf, O., & Welzl, E. (1991). Euclidean minimum spanning trees and bichromatic closest pairs. Discrete & Computational Geometry, 6(3), 407-422.

[5] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications magazine, 40(8), 102-114

[6] Ali, S., Qaisar, S. B., Saeed, H., Khan, M. F., Naeem, M., & Anpalagan, A. (2015). Network challenges for cyber physical systems with tiny wireless devices: a case study on reliable pipeline condition monitoring. Sensors, 15(4), 7172-7205.

[7] Allavena, A., Demers, A., & Hopcroft, J. E. (2005, July). Correctness of a gossip based membership protocol. In Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing (pp. 292-301). ACM.

[8] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. IEEE Communications Surveys & Tutorials, 16(4), 1996-2018.

[9] ANSI/IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ANSI/IEEE std 802.11, 1999.

[10] Aoyang Zhao, Tie Qiu , Feng Xia, Chi Lin, Diansong Luo .A Scale-Free Network Model for Wireless Sensor Networks in 3D Terrain ,Industrial IoT Technologies and Applications ,Volume 173 of the series Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering pp 201-210 , copyright 2016.

[11] Arya, S., & Mount, D. M. (1993, January). Approximate Nearest Neighbor Queries in Fixed Dimensions. In SODA (Vol. 93, pp. 271-280).

[12] Aslam, N.; Phillips, W.; Robertson, W.; Sivakumar, S. A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks. Inf. Fusion. 2011, 12, 202–212.

[13] Atay, F.; Stojmenovic, I.; Yanikomeroglu, H. Generating Random Graphs for the Simulation of Wireless Ad Hoc, Actuator, Sensor, and Internet Networks. In Proc. 8th IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks WoWMoM, Helsinki, Finland, June 2007

[14] Aysal, T. C., Yildiz, M. E., Sarwate, A. D., & Scaglione, A. (2009). Broadcast gossip algorithms for consensus.IEEE Transactions on Signal processing, 57(7), 2748-2761

[15] Barbancho J, Leon .C, Molina F.J. , Barbancho .A (2007) "Using Artificial Intelligence in routing Scheme for wireless Netwoks" Computer Communication 30, Elsevier, pp, 2802-2811.

[16] Bari, A., Jaekel, A., Jiang, J., & Xu, Y. (2012). Design of fault tolerant wireless sensor networks satisfying survivability and lifetime requirements. Computer Communications, 35(3), 320-333.

[17] Belding-Royer, E. M., & Perkins, C. E. (2003). Evolution and future directions of the ad hoc on-demand distance-vector routing protocol. Ad Hoc Networks, 1(1), 125-150.D33,

[18] Blough, D. M., Leoncini, M., Resta, G., & Santi, P. (2006). The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. IEEE Transactions on Mobile Computing, 5(9), 1267-1282.

[19] Bollob´as, B. Random Graphs, 2nd ed.; Cambridge University Press: Cambridge, UK, 2001.

[20] Boyinbode, O., Le, H., & Takizawa, M. (2011). A survey on clustering algorithms for wireless sensor networks. International Journal of Space-Based and Situated Computing, 1(2-3), 130-136.

[21] Bullmore, E., & Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. Nature Reviews Neuroscience, 10(3), 186-198.

[22] F. Chou, J. J. Su, and C. Y. Chen, "Straight Line Routing for wireless sensor networks," in Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC '05),110–115, IEEE, June 2005.

[23] Farahani, S. (2011). ZigBee wireless networks and transceivers. Newnes.

[24] K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," IEEE Communications Magazine, vol. 39, no. 6, pp. 138–147, 2001.

[25] Moallemi and B. V. Roy, "Consensus propagation," IEEE Trans.Inf. Theory, vol. 52, pp. 4753–4766,Nov. 2006.

[26] Caldarelli, G. (2007). Scale-free networks: complex webs in nature and technology. Oxford University Press.

[27] Carsten Moldenhauer "Primal-Dual Approximation Algorithms for Node Weighted Steiner Forest on Planar Graphs" ICALP 2011, Part I,LNCS 6755, pp. 748– 759, 2011. Springer-Verlag Berlin Heidelberg 2011.

[28] Chamam, A., & Pierre, S. (2010). A distributed energy-efficient clustering protocol for wireless sensor networks. Computers & electrical engineering, 36(2), 303-312.

[29] Chen, B., Jamieson, K., Balakrishnan, H., & Morris, R. (2002). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. Wireless networks, 8(5), 481-494.

[30] Chiranjib Patra, Samiran Chattopadhyay, Matangini Chattopadhyay, and Parama Bhaumik, Analysing Topology Control Protocols in Wireless Sensor Network Using Network Evolution Model, International Journal of Distributed Sensor Networks, vol. 2015, Article ID 693602, 8 pages, 2015. doi:10.1155/2015/693602.L2,

[31] Chiranjib Patra. "Using Kohonen's Self-Organizing Map for Clustering in Sensor Networks", International Journal of Computer Applications 1(24):80–81, February 2010. Published By Foundation of Computer Science.

[32] Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," Proc. IEEE, vol. 91, no. 8, pp. 1247–1256, Aug.2003.

[33] Clad, F., Gallais, A., & Mérindol, P. (2012, June). Energy-efficient data collection in wsn: A sink-oriented dynamic backbone. In Communications (ICC), 2012 IEEE International Conference on (pp. 276-280). IEEE.

[34] Costa, L. D. F., Oliveira Jr, O. N., Travieso, G., Rodrigues, F. A., Villas Boas, P. R., Antiqueira, L., ... & Correa Rocha, L. E. (2011). Analyzing and modeling real-world phenomena with complex networks: a survey of applications. Advances in Physics, 60(3), 329-412.

[35] Crossbow, "MICA, Wireless Measurement System Datasheet," 2009, http://stomach.v2.nl/docs/Hardware/DataSheets/Sensors/ MICA data sheet.pdf.

[36] D.W. Gage, Command control for many-robot systems, in: Nineteenth Annual AUVS Technical Symposium, Reprinted in Unmanned Systems Magazine 10 (4) (1992) 28–34.

[37] Deb, B., Bhatnagar, S., & Nath, B. (2002). A topology discovery algorithm for sensor networks with applications to network management.

[38] Deb, B., Bhatnagar, S., & Nath, B. (2004). STREAM: Sensor topology retrieval at multiple resolutions. Telecommunication Systems, 26(2), 285-320.

[39] Deb, S. Bhatnagar and B. Nath. "A topology discovery algorithm for sensor networks with applications to network management". Technical Report dcs-tr-441. Rutgers University. May 2001.

[40] Diestel, R. Graph Theory, electronic edition; Springer-Verlag: Heidelberg, Germany, 2005

[41] Dowell, L.J.; Bruno, M.L. Connectivity of Random Graphs and Mobile Networks: Validation of Monte Carlo Simulation Results. In Proc. the 2001 ACM symposium on Applied computing (SAC ’01), ACM, New York, NY, USA, March 2001

[42] Perkins. Ad Hoc Networking. Addison-Wesley, 2000.

[43] Erd˝os, P.; R´enyi, A. On Random Graphs. Publ. Math. 1959, 6, 290–297

[44] Erd˝os, P.; R´enyi, A. On the Evolution of Random Graphs. Publ. Math. Inst. Hungarian Acad.Sci. 1960, 5, 17–61.

[45] Ergen, S. C., & Varaiya, P. (2006). PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks. IEEE Transactions on Mobile Computing, 5(7), 920-930.

[46] Evangelatos, O., Samarasinghe, K., & Rolim, J. (2013, May). Syndesi: A framework for creating personalized smart environments using wireless sensor networks. In Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on (pp. 325-330). IEEE.

[47] Fan Z P, Chen G R, Zhang Y N. “A comprehensive multi-local-world model for complex networks”. Phys. Lett. A, 2009, 373, 1601–1605

[48] Fang, Q., Gao, J., Guibas, L. J., De Silva, V., & Zhang, L. (2005, March). GLIDER: Gradient landmark-based distributed routing for sensor networks. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE (Vol. 1, pp. 339-350). IEEE.

[49] Federer, S., Wang, Y., & Gao, J. (2009). Connectivity-based localization of large-scale sensor networks with complex shape. ACM Transactions on Sensor Networks (TOSN), 5(4), 31.

[50] Frank Comeau., Nauman Aslam. ,Analysis of LEACH Energy Parameters,Procedia Computer Science ,Volume 5, 2011, Pages 933-938 ,The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011).

[51] Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, “Simulation based optimization of communication protocols for large-scale wireless sensor neworks,” in Proceedings of the IEEE Aerospace Conference, Big Sky, Mont, USA, March, 2003.

[52] Ghosh, A., & Das, S. K. (2008). Coverage and connectivity issues in wireless sensor networks: A survey. Pervasive and Mobile Computing, 4(3), 303-334.

[53] Gkantsidis, C.; Mihail, M.; Zegura, E. The Markov Chain Simulation Method for Generating Connected Power Law Random Graphs. In Proc. 5th SIAM Workshop on Algorithm Engineering and Experiments (ALENEX), Baltimore, MA, USA, January 2003.

[54] Goyal, D., & Tripathy, M. R. (2012, January). Routing protocols in wireless sensor networks: a survey. In Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on (pp. 474-480). IEEE.

[55] Guan Z H, Wu Z P. “The physical position neighbourhood evolving network model”, Physica A, 2007, 387, 314 – 322.

[56] Guo, S., He, L., Gu, Y., Jiang, B., & He, T. (2014). Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. IEEE Transactions on Computers, 63(11), 2787-2802.

[57] Guo, S., Kim, S. M., Zhu, T., Gu, Y., & He, T. (2011, October). Correlated flooding in low-duty-cycle wireless sensor networks. In Network Protocols (ICNP), 2011 19th IEEE International Conference on (pp. 383-392). IEEE.

[58] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on wireless communications, 1(4), 660-670.D56,H28,

[59] Hong, X., & Liang, Q. (2004, September). An access-based energy efficient clustering protocol for ad hoc wireless sensor network. In Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on (Vol. 2, pp. 1022-1026). IEEE.

[60] Hou, T. C., & Tsai, T. J. (2001). A access-based clustering protocol for multihop wireless ad hoc networks. IEEE journal on selected areas in communications, 19(7), 1201-1210.

[61] http://www.csee.usf.edu/~labrador/Atarraya.

[62] Huang, H. (2007). Distributed Computing in Wireless Sensor Networks. In Encyclopedia of Mobile Computing and Commerce (pp. 202-206). IGI Global.

[63] Stojmenovic and X. Lin, “Power-aware localized routing in wireless networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 11, pp. 1122–1133, 2001.

[64] I.F. Akyildiz et al,'Wireless sensor networks: a survey', Computer Networks 38 (2002) 393-422.

[65] J. C. Haartsen and S. Mattisson. Bluetooth -A New Low-Power Radio Interface Providing Short-Range Connectivity. Proceedings of the IEEE, October 2000.

[66] J. F. Yan and Y. L. Liu, "Improved LEACH routing protocol for large scale wireless sensor networks routing," 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, 2011, pp. 3754-3757.

[67] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K.Pister. System architecture directions for networked sensors. In Proceedings of the 9th ACM International Conference on Architectural

Support for Programming Languages and Operating Systems, pages 93-104, Cambridge, Massachusetts, Nov. 2000.

[68] J. Hong, J. Cao, W. Li, S. Lu, and D. Chen, “Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks,” IEEE Transactions on Vehicular Technology, vol. 59, no. 1, pp. 307–318, 2010 11

[69] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, “Minimizing delay and maximizing lifetime for wireless sensor networks with anycast,” IEEE/ACM Transactions on Networking, vol. 18, pp. 515–528, 2010 3.

[70] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proc. ACM SenSys, 2004, pp. 95–107.5.

[71] J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks”, Nature, vol. 393, no.6684, pp. 440 – 442, 1998

[72] J. Wu, M. Cardei, F. Dai, and S. Yang. Extended dominating set and its applications in ad hoc networks using cooperative communication, IEEE Trans. on Parallel and Distributed Systems, 17(8):851–864, 2006.

[73] Jan, M. A., Nanda, P., & He, X. (2013, June). Energy evaluation model for an improved centralized clustering hierarchical algorithm in WSN. In International Conference on Wired/Wireless Internet Communication (pp. 154-167). Springer, Berlin, Heidelberg.

[74] Jardosh, S., & Ranjan, P. (2008, January). A survey: topology control for wireless sensor networks. In Signal Processing, Communications and Networking, 2008. ICSCN'08. International Conference on (pp. 422-427). IEEE.

[75] Jelasity, M., & Babaoglu, O. (2005). T-Man: Gossip-based overlay topology management. Engineering Self-Organising Systems, 3910, 1-15.

[76] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal ,Wireless sensor network survey, Computer Networks 52 (2008) 2292–2330.

[77] K. Han, Y. Liu, and J. Luo, “Duty-cycle-aware minimum energy multicasting in wireless sensor networks,” IEEE/ACM Transactions on Networking, 2012,15

[78] K. Holger and A. Willig, Protocols and Arhitectures for Wireless Sensor Networks. John Wiley and Sons, 2005

[79] K. Naveen and A. Kumar, “Tunable locally-optimal geographical forwarding in wireless sensor networks with sleep-wake cycling nodes,” in Proc. IEEE INFOCOM, 2010, pp. 920–928.8.

[80] Kempe, A. Dobra, and J. Gehrke, “Computing aggregate information using gossip,” in Proc. Foundations of Computer Science, Cambridge, MA, Oct. 2003

[81] Keshri, N., Gupta, A., & Mishra, B. K. (2016). Impact of reduced scale free network on wireless sensor network. Physica A: Statistical Mechanics and its Applications, 463, 236-245.

[82] Kyung-Joon Park, LaeYoung Kim, and Jennifer C. Hou"Adaptive Physical Carrier Sense in Topology-Controlled Wireless Network" .IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 9, NO.1, JANUARY 2009.

[83] L. Barab´asi, R. Albert, and H. Jeong, “Mean-field theory for scale-free random networks,” Physica A, vol. 272, no. 1, pp. 173–187, 1999

[84] L. Cai, H. Huang, S. Blackshaw, J. S. Liu, C. Cepko, and W. H. Wong, “Clustering analysis of SAGE data using a Poisson approach,” Genome Biology, vol. 5, no. 7, p. R51, 2004

[85] L. He, “Energy-effecient multi-path routing woth short latency and low overhead for wireless sensor networks,” in Proceedings of the ACIS International Conference on Software Engineer-ing, Artificial Intelligence, Networking, and Parallel/Distributed Computing, vol. 3, pp. 161–167, 2007.

[86] L. J. Chen, D. X. Chen, X. Li, and J. N. Cao, “Evolution of wireless sensor network”, Proceedings of the IEEE Wireless communications and Networking Conference (WCNC ’07), vol. 556, pp. 3005–3009,

[87] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number," in Conference Record, National Telecommunications Conference,Birmingham, Alabama, December 1978, p. 4.3.2–4.3.5.

[88] L. Su, B. Ding, Y. Yang, T. F. Abdelzaher, G. Cao, and J. C. Hou,“Ocast: optimal multicast routing protocol for wireless sensor networks,”in Proc. IEEE ICNP, 2009, pp. 151–160 14

[89] L. Zhu, H. Luo, H. P. Peng, L. X. Li, and Q. Luo, “Complex networks-based nergy-efficient evolution model for wireless sensor networks”, Chaos, Solitons and Fractals, vol. 41, no. 4, pp. 1828–1835, 2009

[90] Li, J., Andrew, L. L., Foh, C. H., Zukerman, M., & Chen, H. H. (2009). Connectivity, coverage and placement in wireless sensor networks. Sensors, 9(10), 7664-7693.

[91] Li, J., Mirkovic, J., Wang, M., Reiher, P., & Zhang, L. (2002, June). SAVE: Source address validity enforcement protocol. In INFOCOM 2002. Twenty-First Annual Joint7 Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 3, pp. 1557-1566). IEEE.

[92] Li, J., Mirkovic, J., Wang, M., Reiher, P., & Zhang, L. (2002, June). SAVE: Source address validity enforcement protocol. In INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 3, pp. 1557-1566). IEEE.

[93] Li, N., & Hou, J. C. (2004, March). Topology control in heterogeneous wireless networks: Problems and solutions. In INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies (Vol. 1). IEEE.

[94] Li, N., Hou, J. C., & Sha, L. (2005). Design and analysis of an MST-based topology control algorithm. IEEE Transactions on Wireless Communications, 4(3), 1195-1206.

[95] Lili Wang, Jianxun Dang, Yi Jin and Huihua Jin, "Scale-free topology for large-scale wireless sensor networks," 2007 3rd IEEE/IFIP International Conference in Central Asia on Internet, Tashkent, 2007, pp. 1-5.

[96] Lindsey, S., & Raghavendra, C. S. (2002). PEGASIS: Power-efficient gathering in sensor information systems. In Aerospace conference proceedings, 2002. IEEE (Vol. 3, pp. 3-3). IEEE.

[97] Lindsey, S., Raghavendra, C. S., & Sivalingam, K. M. (2001, April). Data Gathering in Sensor Networks using the Energy Delay Metric. In IPDPS (Vol. 188).

[98] Liu, H., Yin, R., Liu, B., & Li, Y. (2016). A scale-free topology model with fault-tolerance and intrusion-tolerance in wireless sensor networks. Computers & Electrical Engineering, 56, 533-543.

[99] Liu, X. (2012). A survey on clustering routing protocols in wireless sensor networks. sensors, 12(8), 11113-11153.

[100] Liu, X., Cao, J., Lai, S., Yang, C., Wu, H., & Xu, Y. L. (2011, April). Energy efficient clustering for WSN-based structural health monitoring. In INFOCOM, 2011 Proceedings IEEE (pp. 2768-2776). IEEE.

[101] Liu, Y., & Wang, Z. (2012). Maximizing energy utilization routing scheme in wireless sensor networks based on minimum hops algorithm. Computers & Electrical Engineering, 38(3), 703-721.

[102] Lurong Jiang, Xinyu Jin, Yongxiang Xia, Bo Ouyang, Duanpo Wu, Xi Chen.nternational Journal of Distributed Sensor Networks Vol 10, Issue 8 A Scale-Free Topology Construction Model for Wireless Sensor Networks ,August-27-2014,doi:10.1155/2014/764698.

[103] M. K. Marina and S. R. Das, “On-demand multipath distance vector routing in ad hoc networks,” in Proceedings of the 19th International Conference on Network Protocols (ICNP ’01),14–23, November 2001

[104] M. Tong and M. Tang, "LEACH-B: An Improved LEACH Protocol for Wireless Sensor Network," 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, 2010, pp. 1-4.

[105] Ma, C., Liu, H., Zuo, D., Wu, Z., & Yang, X. (2011). Research on key nodes of wireless sensor network based on complex network theory. Journal of Electronics (China), 28(3), 396-401.

[106] Meguerdichian, S., Slijepcevic, S., Karayan, V., & Potkonjak, M. (2001, October). Localized algorithms in wireless ad-hoc networks: location discovery and sensor exposure. In Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing (pp. 106-116). ACM.

[107] Muruganathan, S. D., Ma, D. C., Bhasin, R. I., & Fapojuwo, A. O. (2005). A centralized energy-efficient routing protocol for wireless sensor networks. IEEE Communications Magazine, 43(3), S8-13.D22,

[108] Neda Enami, Reza Askari Moghadam .Energy Based Clustering Self Organizing Map Protocol For extending Wireless Sensor Networks lifetime and coverage ,Canadian Journal on Multimedia and Wireless Networks Vol. 1, No. 4, August 2010.

[109] Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, “On distributed averaging algorithms and quantization effects,” MIT, Cambridge, MA, LIDS Rep. 2274, 2007.

[110] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks,” in Proc. USENIX/ACM Symp. On Networked Systems Design and Implementation, vol. 246, 2004K.

[111] P. M. Wightman. Atarraya: A topology control simulator. http://www.cse.usf.edu/˜pedrow/atarraya/.

[112] P. T. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massouli´ e. Epidemic information dissemination in distributed systems. IEEE Computer, 37(5):60–67,2004.

[113] Pan, L., Wu, H., & Tzeng, N. F. (2009, June). An efficient and scalable prioritized MAC protocol (PMAC) for backbone communication in wireless sensor networks. In Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on (pp. 508-513). IEEE.

[114] Patra, C., Mondal, A., Bhaumik, P., & Chattopadhyay, M. (2012). Topology Management in Wireless Sensor Networks. In Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management (pp. 14-24). IGI Global.

[115] Patra, C., Mondal, A., Bhaumik, P., & Chattopadhyay, M. (2012). An energy efficient event based hierarchical clustering algorithm for wireless sensor networks. Global Trends in Computing and Communication Systems, 380-385

[116] Patra.C and Botezatu .N, "Effect of gossiping on some basic wireless sensor network protocols," 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, 2017, pp. 309-314

[117] Patra. C, Botezatu. N,Improving the Performance of Hierarchical Clustering Protocols with Network Evolution Model, Advances in Intelligent Systems and Computing, Springer, ISSN 2194-5357, 2018

[118] Patra, C., Chattopadhyay, M., Bhaumik, P., & Roy, A. G. (2011, February). Using self organizing map in wireless sensor network for designing energy efficient topologies. In Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on (pp. 1-6). IEEE

[119] Patra, C., Chattopadhyay, M., Bhaumik, P., Bhattacharya, M., & Mukherjee, S. (2014, February). A reliable two-tier energy-efficient topology building algorithm for Wireless Sensor Networks. In Applications and Innovations in Mobile Computing (AIMoC), 2014 (pp. 146-150). IEEE.

[120] Patra, C., Guha Roy, A., Chattopadhyay, S., & Bhaumik, P. (2010). Designing energy-efficient topologies for wireless sensor network: neural approach. International Journal of Distributed Sensor Networks, 6(1), 216716.

[121] Pedro Wightman and Miguel A. Labrador, “A3: A Topology Control Algorithm for Wireless Sensor Networks,” IEEE Globecom,November 2008.

[122] Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.

[123] Petrovic, R. C. Shah, K. Ramchandran, and J. Rabaey, “Data Funneling: Routing with aggregation and compression for wireless sensor networks,” in Proceedings of IEEE Sensor Network Protocols and Applications (SNPA), 2003.

[124] Qiu, T., Zhao, A., Xia, F., Si, W., & Wu, D. O. (2017). ROSE: Robustness strategy for scale-free wireless sensor networks. IEEE/ACM Transactions on Networking, 25(5), 2944-2959.

[125] R. Chandra, V. Ramasubramanian, and K. Birman. Anonymous gossip:Improving multicast reliability in mobile ad-hoc networks. In Proc. 21st International Conference on Distributed Computing Systems (ICDCS),pages 275–283, 2001.

[126] R. Gallagher, T. Appenzeller, “Beyond Reductionism” , Science, 1999, Vol. 284 no. 5411 pp. 79 .

[127] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, “Randomized rumor spreading,” in Annual Symp. on Foundations of Computer Science, vol. 41, 2000, pp. 565–574.

[128] R. Ramanathan and R. R. Hain. Topology Control of Multihop Wireless Networks using Transmit Power Adjustment. In IEEE INFOCOM,pp. 404–413, 2000.

[129] R. Wattenhofer And A. Zollinger. “XTC: A practical topology control algorithm for ad-hoc Networks ”. Proceedings of the 18th international parallel and distributed processing Symposium, 2004. 26-30 April 2004 page(s):216.

[130] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang .Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Net-works. In IEEE INFOCOM, pages 1388–1397, 2001

[131] R.F. Service, “Complex systems. Exploring the systems of life”. Science. 1999, Vol 284 no. 5411 pp. 83.

[132] Rajendra Prasad Mahapatra, Rakesh Kumar Yadav ,Descendant of LEACH Based Routing Protocols in Wireless Sensor Networks, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), Procedia Computer Science 57 (2015) 1005 – 1014.

[133] Ramos, H. S., Boukerche, A., Oliveira, A. L., Frery, A. C., Oliveira, E. M., & Loureiro, A. A. (2016). On the deployment of large-scale wireless sensor networks considering the energy hole problem. Computer Networks, 110, 154-167.

[134] Razaque, M. Abdulgader, C. Joshi, F. Amsaad and M. Chauhan, "P-LEACH: Energy efficient routing protocol for Wireless Sensor Networks," 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, 2016, pp. 1-5.

[135] S. C. Zhang, P. K. Wong, D. G. Sachs, R. Koetter, and D.L. Jones, “Stability analysis of stochastic sensor networks,” in Proceedings of the Asilomar Conference, 2004.

[136] S. D. Li, L. X. Li, and Y. X. Yang, “A local-world heterogeneous model of wireless sensor networks with node and link diversity,” Physica A, vol. 390, no. 6, pp. 1182–1191, 2011.

[137] S. Guha, P. Basu, C.-K. Chau, and R. Gibbens, “Green wave sleep scheduling: Optimizing latency and throughput in duty cycling wireless networks,” IEEE Journal on Selected Areas in Communications, vol. 29,no. 8, pp. 1595–1604, 2011.

[138] S. Guo, Y. Gu, B. Jiang, and T. He, “Opportunistic flooding in low-duty cycle wireless sensor networks with unreliable links,” in Proc. ACM MobiCom, 2009, pp. 133–144.9

[139] S. Haykin, Neural Networks A comprehensive Foundation, Pearson Education, New Delhi, India, 2007.

[140] S. J. Lee and M. Gerla, “Split multipath routing with maximally disjoint paths in Ad hoc networks,” in Proceedings of the International Conference on Communications (ICC ’01),3201–3205, 2001.

[141] S. Lim, C. Yu and C. R. Das, "Clustered Mobility Model for Scale-Free Wireless Networks," Proceedings. 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, 2006, pp. 231-238.doi: 10.1109/LCN.2006.322105.

[142] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu. The broadcast storm problem in a mobile ad hoc network. In Proc. fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 151–162, 1999.

[143] S.Boyd, A.Ghosh, B.Prabhakar and D.Shah, “Randomized gossip algorithms,” IEEE Trans. Inf. Theory, vol. 52,pp. 2508–2530, Jun. 2006.

[144] S.D. Li, L. X. Li, and Y. X. Yang, “A local-world heterogeneous model of wireless sensor networks with node and link diversity”, Physica A, vol. 390, no. 6, pp. 1182–1191, 2011

[145] Savarese, C., Rabaey, J. M., & Beutel, J. (2001). Location in distributed ad-hoc wireless sensor networks. In Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on (Vol. 4, pp. 2037-2040). IEEE.

[146] Shahbazi, H., Araghizadeh, M.A., Dalvi, M., (2008) “Minimum Power Intelligent Routing In Wireless Sensors Networks Using Self Organizing Neural Networks”, IEEE International Symposium on Telecommunications, pp. 354—358.

[147] Sheridan P, Yagahara Y, Shimodaira H. “A preferential attachment model with Poisson growth for scale-free networks”, Annals of the institute of statistical mathematics, 2008, 60, 747 – 761.

[148] Sudhanshu Tyagi, , Neeraj Kumar, A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks ,Journal of Network and Computer Applications,Volume 36, Issue 2, March 2013, Pages 623 645.

[149] Sun, Y., Gurewitz, O., Du, S., Tang, L., & Johnson, D. B. (2009, November). ADB: an efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (pp. 43-56). ACM.

[150] Tran, D. A., & Nguyen, T. (2008). Localization in wireless sensor networks based on support vector machines. IEEE Transactions on Parallel and Distributed Systems, 19(7), 981-994.

[151] U. Colesanti, S. Santini, and A. Vitaletti, “Dissense: an adaptive ultralow-power communication protocol for wireless sensor networks,” in Proc. Distributed Computing in Sensor Systems (DCOSS), 2011, pp.1–10.

[152] Üster, H., & Lin, H. (2011). Integrated topology control and routing in wireless sensor networks for prolonged network lifetime. Ad Hoc Networks, 9(5), 835-851.

[153] Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Duke Technical Report CS-2000-06, July 2000.

[154] Vesanto J, Alhoniemi E. (2000) “Clustering of Self Organizing Map”.IEEE Transactions on Neural Networks, Vol.11, No 3, pp.586-358

[155] Viani, F., Salucci, M., Rocca, P., Oliveri, G., & Massa, A. (2012, March). A multi-sensor wsn backbone for museum monitoring and surveillance. In Antennas and Propagation (EUCAP), 2012 6th European Conference on (pp. 51-52). IEEE.

[156] W. Ren and R. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” IEEE Trans.Autom. Control, vol. 50, pp. 655–661, 2005

[157] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless wensor networks,” in Proc. IEEE INFOCOM, 2002, pp.1567–1576.2.

[158] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan An application-specific protocol architecture for wireless microsensor networks IEEE Transactions on Wireless Communications, 1 (4) (2002), pp. 660 670

[159] Wang L N, Guo J L, Yang H X, Zhou T. “Local preferential attachment model for hierarchical networks”, Physica A, 2009, 388, 1713-1720.

[160] Wang, H. Jin, J. Dang and Y. Jin, "A Fault Tolerant Topology Control Algorithm for Large-Scale Sensor Networks," Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007), Adelaide, SA, 2007, pp. 407-412.doi: 10.1109/PDCAT.2007.31

[161] Wendi Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, Energy-Efficient Communication Protocols for Wireless Microsensor Networks, Proc. Hawaaian Int'l Conf. on Systems Science, January 2000.

[162] Wightman, P. M., & Labrador, M. A. (2011, March). A3Cov: a new topology construction protocol for connected area coverage in WSN. In Wireless Communications and Networking Conference (WCNC), 2011 IEEE (pp. 522-527). IEEE.

[163] Wightman, P.M., Labrador, "A3: A Topology Construction Algorithm for Wireless Sensor Networks", M.A., Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.

[164] http://www.wireless-tag.com/upload/file/p1ank0n1cipvu1rsb1upm16j91d323.pdf

[165] X. Jiao, W. Lou, J. Ma, J. Cao, X. Wang, and X. Zhou, “Duty-cycle aware minimum latency broadcast scheduling in multi-hop wireless networks,” in Proc. IEEE ICDCS, 2010, pp. 754–763.13

[166] X. Li and G. Chen, “A local-world evolving network model,”Physica A, vol. 328, no. 1-2, pp. 274–286, 2003.

[167] X. Li and G. Chen, A local-world evolving network model,Physica A,281 vol. 328, no. 1-2, pp. 274286, 2003.

[168] X. Li and G. R. Chen, “A local-world evolving network model”, Physica A, vol. 328, no. 1-2, pp. 274–286, 2003.

[169] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, Proceedings of the 1st international conference on embedded networked sensor systems, ser. SenSys'03, ACM, New York, NY, USA (2003), pp. 28-39.

[170] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, Proceedings of the 1st international conference on

embedded networked sensor systems, ser. SenSys'03, ACM, New York, NY, USA (2003), pp. 28-39.

[171] Xiaojuan Luo, Huiqun Yu, and Xiang Wang, “Energy-Aware Topology Evolution Model with Link and Node Deletion in Wireless Sensor Networks”, Mathematical Problems in Engineering, vol. 2012, Article ID 281465, 2012. doi:10.1155/2012/281465

[172] Y. C. Tseng, C.-S. Hsu, and T. Y. Hsieh. Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. In IEEE INFOCOM,2002.

[173] Y. Gu and T. He, “Dynamic switching-based data forwarding for low duty-cycle wireless sensor networks,” IEEE Transactions on Mobile Computing, vol. 10, no. 12, pp. 1741–1754, 2011.4

[174] Y. Sun, O. Gurewitz, S. Du, L. Tang, and D. B. Johnson, “Adb:an efficient multihop broadcast protocol based on asynchronous dutycycling in wireless sensor networks,” in Proc. ACM SenSys, 2009, pp.43–56.

[175] Yang and I. Aug-Blum, “Delivery ratio-maximized wakeup scheduling for ultra-low duty-cycled wsns under real-time constraints,” Computer Networks, vol. 55, no. 3, pp. 497–513, 2011.7

[176] Yassein, M. B., Khamayseh, Y., & Mardini, W. (2009, June). Improvement on LEACH protocol of wireless sensor network (VLEACH. In Int. J. Digit. Content Technol. Appl. 2009.

[177] Ye, M.; Li, C.; Chen, G.; Wu, J. An energy efficient clustering scheme in wireless sensor networks. Ad Hoc Sens. Wirel. Netw. 2006, 3, 99–119.

[178] Ye, M.; Li, C.; Chen, G.; Wu, J. EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks. Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC), Phoenix, AZ, USA, 7– 9 April 2005; pp. 535–540.

[179] Younis, O., & Fahmy, S. (2004, March). Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (Vol. 1). IEEE.

[180] Younis, O.; Fahmy, S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. IEEE Trans. Mobile Comput. 2004, 3, 366–379

[181] YU. Wang, W. Wang, and X. Y. Li, “Effecient distributed low-cost backbone formation for wireless networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 7.

[182] Yuanyuan, Z., Jia, X., Yanxiang, H.: Energy efficient distributed connected dominating sets construction in wireless sensor networks.In: Proceeding of the 2006 ACM International Conference on Communications and Mobile Computing, pp. 797–802 (2006).

[183] Yue Wang, Erwu Liu, Xiaojun Zheng, Zhengqing Zhang, Yuhui Jian, Xuefeng Yin, and Fuqiang Liu Energy-aware complex network model with compensation 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)Lyon, 2013, pp. 472-476.doi: 10.1109/WiMOB.2013.6673401.

[184] Z. Li, Y. Peng, D. Qiao, and W. Zhang, "Lba: lifetime balanced data aggregation in low duty cycle sensor networks," in Proc. IEEE INFOCOM, 2012, pp. 1844 –1852.

[185] Z.-H. Guan and Z.-P. Wu, "The physical position neighbour-hood evolving network model," Physica A, vol. 387, no. 1, pp.314–322, 2008.

[186] Zhang, S., Er, M. J., Zhang, B., & Naderahmadian, Y. (2017). A novel heuristic algorithm for node localization in anisotropic wireless sensor networks with holes. Signal Processing, 138, 27-34.

[187] Zhang, X. (2009, September). Model design of wireless sensor network based on scale-free network theory. In Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on (pp. 1-4). IEEE.

[188] Zhao, J., & Govindan, R. (2003, November). Understanding packet delivery performance in dense wireless sensor networks. In Proceedings of the 1st international conference on Embedded networked sensor systems (pp. 1-13). ACM.

[189] Zheng, G., & Liu, Q. (2013). Scale-free topology evolution for wireless sensor networks. Computers & Electrical Engineering, 39(6), 1779-1788.

[190] Zheng, G., Liu, S., & Qi, X. (2012). Scale-free topology evolution for wireless sensor networks with reconstruction mechanism. Computers & Electrical Engineering, 38(3), 643-651.

[191] Zhenyu Zhang et al 'Resource prioritization of code optimization techniques for program synthesis of wireless sensor network applications', Journal of Systems and Software',Volume 82 ,Issue9 ,September 2009.

[192] Zhu, C., Zheng, C., Shu, L., & Han, G. (2012). A survey on coverage and connectivity issues in wireless sensor networks. Journal of Network and Computer Applications, 35(2), 619-632.

[193] Zhu, H., Luo, H., Peng, H., Li, L., & Luo, Q. (2009). Complex networks-based energy-efficient evolution model for wireless sensor networks. Chaos, Solitons & Fractals, 41(4), 1828-1835.

Chapter 1
Introduction

Preamble

In situations like disaster management, battlefield or inhospitable areas monitoring is the main concern, as it helps in taking quick and right decisions. To aid this process there is a need of technology that would allow acquisition of the precious data without any human intervention.

The right technology that fits the requirement is the need for small inexpensive devices which is portable enough for operation from personal area network to very large scale network with limited infrastructure for power and communication is the *wireless sensor networks (WSN).*The functioning of these devices with limited resources are one of the many specialties of these devices which makes them to work in low energy systems. One of the many ways in driving these miniature systems in an efficient way is by Topology Management (TM). Nowadays this concept of topology management is one of the important factors for wireless network construction as it directly affects the energy consumption of the node.

1.1 Wireless Sensor Networks (WSN)

1.1.1 Wireless Sensor Networks Architecture

WSN is a wireless network consisting of autonomously distributed devices using sensors to measure different measurable parameters. A WSN system consists of a gateway that includes wireless connectivity to the distributed nodes (see Figure 1). The selected wireless protocol depends on user's application requirements. The most popular available standards include 2.4 GHz radios based on either IEEE 802.15.4 or IEEE 802.11 (Wi-Fi) standards or proprietary radios, which are usually 900 MHz

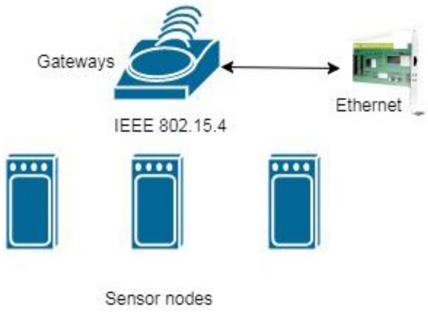


Figure-1.1: WSN System Architecture

A WSN node contains several electronic components. These consists the radio, microcontroller, analog circuit, battery and sensor interface. While using radio technology, the most important trade-offs are in battery-powered systems, higher radio data rates and most frequent radio utilization that consumes greater power. Often 2 years of battery life is a requirement, hence many of the WSN systems are Zig Bee [22] based due to its minimum-power consumption. As battery life and power management technologies are constantly changing over time so the available IEEE 802.11 bandwidth, Wi-Fi is becoming an interesting technology. The dataset below gives the relative energy consumption values of various Tx and Rx modes for IEEE802.11 devices [150] and Zig Bee are given below.

Mode	**Standard**	**Speed rate**	**Typical Value**	**Unit**
Tx	11b	1	215	mA
		11	197	
	11g	6	197	
		54	145	
	11n	MCS7	120	
Rx	All rates		56	mA

Table-1.1 Operating Power Consumption of WT8266-S2 Wi-Fi module

Action	**Duration**	**Average current**	**Energy (mAh)**
Device in Sleep	1 hr	1microAmp	$1X10^{-3}$
Device transitions to active mode	10 millisec	50 microSec	$1.39X10^{-3}$
device transmits the packets	550 microSec	20 milliAmp	$3.06X10^{-6}$
Device is in receive mode to acknowledgement	400microSec	20milliAmps	$2.22X10^{-6}$

Table-1.2 Typical power consumption for Zig bee transceiver

To enhance the lifetime of the battery, a WSN node periodically wakes up and sleep, the transmission of data packets is done in wake up mode and sleep mode is merely used for energy conservation purpose. This means that radio technology must be good enough to transmit the data signal and allow the system back to sleep with minimum power. So the processor involved must be able to wake up with power on, and go back to sleep state efficiently. Microprocessor trends for WSNs include reducing power consumption while maintaining or increasing processor speed. Much like the radio choice, the power consumption and processing speed trade-off is a key concern when selecting a processor for WSNs.

1.1.2 Common topologies in WSN

WSN nodes in network are hierarchically sorted in any one of three types. In a star topology, each node connects directly to a gateway. In a clustered tree network, every node is connected to a node higher in the tree and then to the gateway node as shown in Figure-1.2, and data is sourced from the highest node on the tree to the gateway. Furthermore to increase reliability and efficiency, mesh networks are used to connect as many nodes in the system and route the data through the most efficient path.

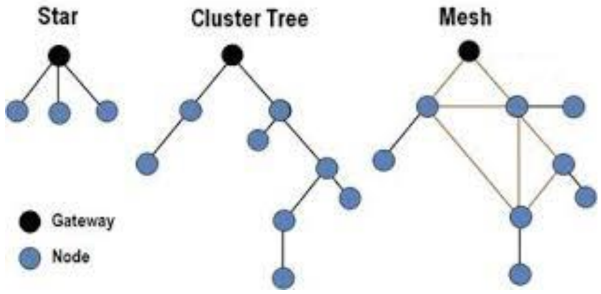


Figure-1.2 Common topologies in WSN

As it can be seen from Figure-1.2, a typical structure of a WSN includes two types of wireless devices: sink nodes and normal nodes.

The sink nodes form the gateway of the WSNs, every data that is generated from the sensor network will be aggregated at the sink node and send to the control site using a second communication card, like cellular, Ethernet or another wireless network. Moreover, the sink nodes use the information from external backbone networks into the WSN, like commands or queries using a software interface between the WSN and the computing device. In addition, the sink nodes keep the track of the energy state and device IDs of the nodes as being the organizational head of the sub-WSN.

The normal nodes which are the majority in network are bound to collect data about the physical variables being observed and reporting it to the sink node. In excess, if the network is so large that some of the sensor nodes are beyond the reach of the sink node directly, the normal nodes use multi hop forwarding schemes to forward the data, so that even the far flanged nodes can communicate their data to the sink node.

Due to the critical operational features of the WSN, the sink nodes should have a better configuration in terms of memory, processing and energy, in contrast to the normal nodes.

Now when the sensor network topology is considered for distributed systems, it uses the following three layers to facilitate topology management:

1) The network layer is in charge of providing data forwarding between the different nodes involved in the network. A network protocol, such as the Multicast Ad-hoc On-demand Distance Vector (MAODV) is needed to provide point-to-point communication.

2) The medium layer called the overlay layer which is a virtual layer that builds the event notification service by providing a network of brokers that redirect notifications to the corresponding subscribers.

3) Finally, on the top layer the event-based protocol is implemented.

However, there are some limitations of the distributed techniques

1. When distributed computation is done to transmit data from one point to another for the nodes in network having local information, then sufficient control messages are generated to construct the topology. This control messages consume a significant battery power of the nodes which in turn bring down the network life time.
2. The exchange of information and additional computation required to achieve inter site co-ordination are a form of energy overhead in distributed systems.

So in our work we have used hierarchical based topologies for present scenario as shown in Figure-1.3 [6], regarding the operation of sensor networks the communication bandwidth and energy are significantly more limited than in a tethered network environment so these two areas have become the most soughed research areas. This can be explained with the following two reasons:

1. The battery life of WSN is expected to work for at least 2 months.

2. The wireless channels are inherently error-prone and have time-varying characteristics this make it tough to obtain consistently good performance throughout the operation cycle.

So protocols and algorithms must be designed to contain the above mentioned conditions. To avoid the redundancy or wear out of the nodes in the network, which is deployed in an inaccessible area, the protocol must be time efficient as well as data efficient. These are the typical reasons for research on WSNs has been mostly on the design of energy and computationally-efficient algorithms and protocols. So out of the many provisions utilized to reduce energy consumptions, the most important techniques used in wireless sensor networks is Topology Management.

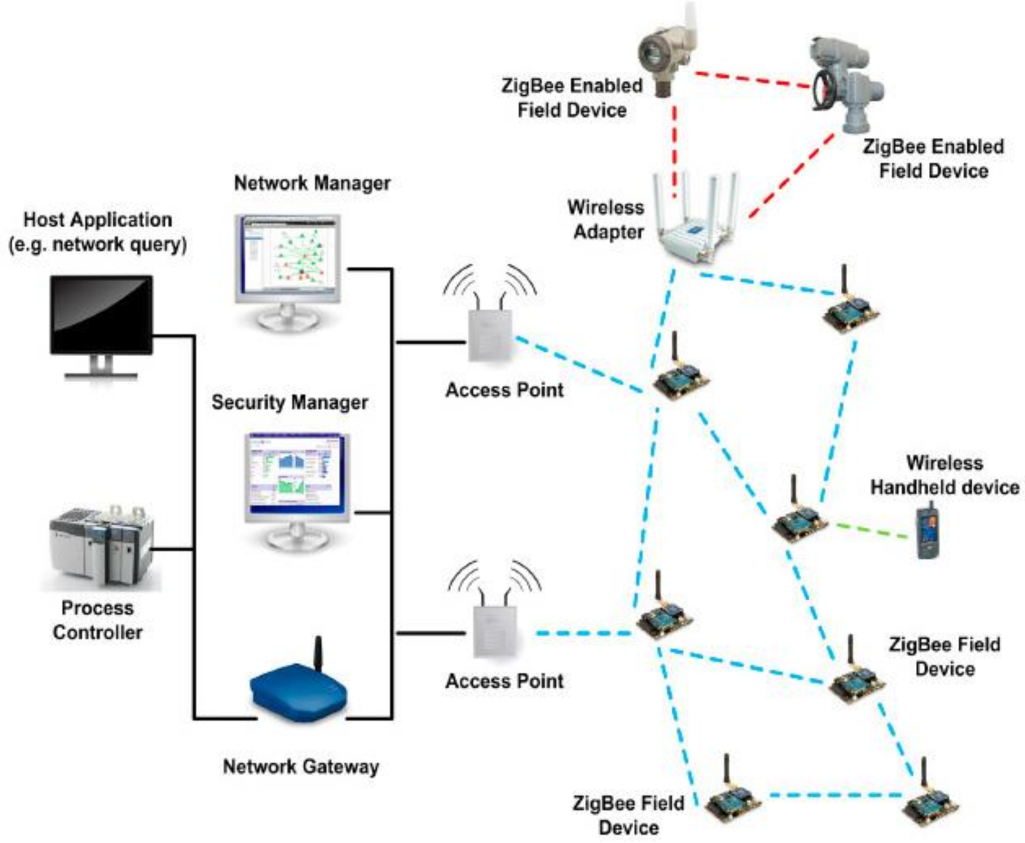


Figure-1.3: A typical sensor network setup with central monitoring

1.2 Topology Management

1.2.1 Network Topology

Before embarking on the understanding of Topology Management, the concepts of network topology should be understood with explicit. To start with the simple definition of network topology as

Network Topology is the set of all active nodes and links in the network along which communication can occur [77].

The definition implies the connectivity between the nodes as one of the important factor for understanding the dynamics of network topology. So from graph theoretic approach one can easily put forward the following assertions.

A network is called connected if this associated graph is connected. A graph G is connected if and only if there exists a path between any pair of its vertices [39]. If a network is connected then any pair of nodes can communicate with each other, possibly taking multiple hops through relay nodes.

However it is clear that the connectivity of a WSN is related to the positions of nodes, and those positions are heavily affected by the method of sensor deployment. To illustrate this fact lets us consider two situations, in the first one the nodes can be placed with desired location and in another case the nodes are dispersed randomly over an area. In the first case the user has fuller control over the topology management, so he knows how and where the nodes be placed so that optimum performance can be expected. But in the second case of node placement which is more challenging actually creates an issues with other problems such as, configure the nodes which are placed randomly that is not suited to give optimal performance. Such topology for WSN can contain all or one of the mentioned characteristics:

1. The sensors may send redundant data to the sink as the sensors are placed randomly.
2. Since there is an uneven distribution of the nodes, so to preserve synchronization there may be some areas which may experience great amount of collisions and thereby delaying the communication of data packets.

3. The above two cases will lead to uneven battery drainage, which may result in unbalanced clustered network.
4. In case of large scale sensor networks the results of performance does not seem to agree with predicted theory.

Thus with respect to above context problems can be modelled with necessary justification in which Topology Management schemes is qualified to resolve.

1.2.2 Scope and Meaning of Topology Management

The main objective of Topology Management is to reduce the energy consumption while keeping the important characteristics of network life time, connectivity and coverage.

In the situation where the user have access to design the topology of the network, the need for topology management is not that important as the user has full discretion on the design issues on the topology of the network. The solution to this kind of problem should encompass the characteristic of the terrain, type, make of the sensors, battery type, routing protocol to name a few. The permutation and combination of these characteristics can lead to the effective management of the total deployment process.

But in case of remote topology deployment, the advantage of instant availability of the network outweighs all the advantages manual deployment. But this advantage comes with the following challenges listed below:

1. The location coordinates of the sensor node cannot be changed.
2. The sleep and wake cycles should be handled intelligently so contain unnecessary data transmission and energy loss.
3. Optimum density of the nodes to be maintained proper monitoring of the desired area. This issue becomes more critical in case of large scale sensor networks.
4. The topology management schemes should be robust enough to encounter the characteristic of the deployment terrain.
5. Due to the advent of wireless charging technology, the charging schedule of the sensor nodes is one of the upcoming research problem.

Hence remote topology deployment has been the center of attraction in WSN research.

In general topology deployment will be implemented successfully if the following steps are followed

1. Topology Management: The topology management provides the backbone to various routing protocols so that there is an energy-efficient communication of the data.
2. Topology Perseverance: In this phase the sensor nodes take in charge of the network ensuring proper data communication and life time of the network.

Over the lifetime of the network operation, it is expected that the above mentioned process will be iterated several times until the network energy is depleted.

1.3 Problem Statement and Contributions

To start with the approaches to solve the problem as described in the above sections, the usage of topology management schemes should have the following characteristics:

1. The algorithm used for topology construction and perseverance should be distributed by nature.
2. These algorithms should have less computational complexity so that the battery power do not exhaust easily.
3. The topology construction algorithm must produce a connected network that will cover the area of interest with a minimum number of nodes
4. While the topology analysis coupled with perseverance algorithms must optimize the network so that the resources of the network are used effectively in enhancing the lifetime of the network.
5. To continue with seamless network performance, maintenance process is indispensable tool to monitor the functioning of the network so that the run time, preventive and predictive maintenance can be performed to avoid break down maintenance.

All the above constraints make the topology management schemes a very challenging problem. This dissertation presents some of the newer topology management schemes which may be described as follows:

1. **Topology Construction**: Preserving the energy or battery power of a wireless sensor network is of major concern. In such types of network, the sensors are deployed in an ad hoc manner, rather than in any deterministic way. Chapter 3 is concerned with applying standard routing protocols into wireless sensor networks by using neural network modified topology, which proves to be energy efficient as compared with unmodified topology.On the contrary of the graph theoretic approach, SOFMTB (Self Organizing Maps Topology Building Algorithm) [117,119], a neural network based concept is used for partitioning the total number of sensor nodes. This scheme allows a certain number of sensor nodes to be active, on condition that the connectivity and the coverage is preserved.

2. **Topology Discovery**: In spite of the various constraints present in wireless sensor networks (WSN), their functionality has become popular across multiple application domains that require the transmission of information. Of these limitations, energy conservation is the most critical aspect, and a known strategies to save energy and to prolong the lifetime of WSNs is topology control. Chapter 4 presents two schemes, a reliable energy-efficient topology control algorithm and a minimum transmission level algorithm, for use in wireless sensor networks. Here topology for clustering has been designed on basis of production rules applied in the context of residual energy and the number of neighbors of the sensor node to produce meaningful cluster head selection. In another method of initialization process, the local information is used to reduce the maximum power to discover the neighboring nodes [118]. Both these methods are found to optimize the consumption of energy in the network.

3. **Network Evolution Model and Application**: The main purpose of network evolution model is to understand the mechanics of clustering in wireless sensor network. This model is applicable for large scale sensor networks [29,116] and is developed using the concept of scale free networks. A large number of algorithms have been devised to compute ‘good’ clusters in a WSN. In this regard network evolution model introduced in Chapter 5

discusses the concept of local world model which is the natural functional mechanism of clustering in wireless sensor networks. This network evolution model is used to evaluate the probability of clustering in (WSN). The application of this network evolution model has been discussed in Chapter 6 where we can find that the life time of the network is improved up to 150 percentage in case of LEACH and LEACH-C protocols. It is also observed that as the number of nodes increases, the better topology management is done and hence the better life time of the WSN. This theoretical framework based on a complex network model is found to have reasonably matched with measures obtained by simulation studies conducted on tree-based clustering algorithms [Chapter 6].

3. **Gossiping Framework**: In wireless sensor networks the message passing from one node to another consumes energy and the message accumulation reduces the performance of the network. In order to mitigate this effect, the usage of gossiping framework has been introduced in Chapter 7 which improves the message complexity of the network and hence save the energy in terms computation and complexity. This method has remarkable success rate of reducing the number of messages by 25% while running with the most widely used protocols like flooding and spanning tree protocol.

1.4 Structure of the Dissertation

The structure of the dissertation is presented as follows. Chapter 2 is the literature review on the issue of topology management which contains detailed discussions on introduction to the proposed taxonomy of topology management to the latest development in this area. Chapter 3 describes a methodology using self-organizing maps from neural networks to partition the network. This partitioning leads to an optimum number of nodes which preserves the connectivity and coverage of the network. Chapter 4 discusses on the usage of production rules using the residual energy and the number of neighbors to determine the number of cluster heads. In this chapter another method of initialization process, the local information is used to reduce the maximum power to discover the neighboring nodes. Both these methods are found to be effective in reducing the energy wastage in the network. Chapter 5 deduces the mathematical framework of clustering for wireless sensor network using network evolution model. This framework works fairly well with existing topology management protocols which has been verified in Chapter 6. In this chapter the concept of network evolution model has been used to calculate the optimum number of cluster heads on the basis of alive nodes in LEACH and LEACH –C. This method has been seen to boost the efficiency of these protocols to 150 percent without much change in the existing algorithm. Chapter 7 implements the usage of gossiping framework to reduce the generation of messages in the network. By using this framework it have been seen that the reduction in the message generation is 25 percent when used on popular routing protocols like spanning tree protocol and flooding protocol. The conclusion and final remarks will be presented in Chapter 8 of this dissertation.

Chapter 2

Literature Review

2.1 Topology Management Taxonomy

One of the primary goals of network management in sensor networks is that it be autonomous. This is especially important in fault and configuration management. Configuration management includes the self-organization and self-configuration of the sensor nodes. Since WSN's involve very little human intervention after deployment, it is imperative that the areas of fault management be self-diagnostic and self-healing. Another important issue to consider in fault management of WSN's is that a single node failure should not impact the operation of the network, unlike a traditional network device failure causing impact to several users to potentially the entire network. There are several new functional areas of network management in sensor networks. Apart from topology management there are still new functional areas introduced for network management of WSN's are energy management and program management. In [64] energy management the most common way to conserve energy in WSN's is to power off a node when idle, but there have been many proposals in existing algorithms and protocols as well as establishing new protocols in order to be more energy efficient.

Program or code management [129] is another aspect of network management in WSN's. The traditional method of updating a program in a sensor node is to attach the node to a programming interface of a laptop or PDA. This is not feasible in many WSN deployments. Transmitting an entire new program version to all sensors in a WSN is not practical as it consumes too much energy and will lead to a short network lifetime. There needs to be a way to transmit minimal packets to all nodes requiring the update while ensuring appropriate nodes receive the update reliably. There have been several proposals in the area of code update/management and it continues to be an active research area.

To begin with topology management [191] there is six properties that should exist in the topology of WSN’s: 1) symmetry, 2) connectivity, 3) spanner, 4) sparseness, 5) low degree, and 6) low interference. It is often observed the case that two properties, connectivity and sparseness conflict with each other. Despite all conflicts, the objective of topology management is to provide a backbone to various routing protocols so that there is an energy-efficient communication of the data. The algorithms responsible for building up the backbone may be categorized into three types (1) topology discovery, (2) sleep cycle management, and (3) clustering.

The following subsections describes about the detailed methods and taxonomy of the above mentioned backbone building algorithms.

2.1.1 Taxonomy of Topology Management for Discovery

In this phase, nodes discover themselves and use their maximum transmission power to build the initial topology.

The Topology Discovery [67, 161] as this phase is popularly called, the algorithm uses the broadcast medium of wireless communication to know the existence of other nodes just by listening to the communication channel. The algorithm takes advantage of the fact to find a set of neighboring nodes using location information to construct the approximate topology of the network. Only neighborhood nodes reply back to the topology discovery messages, thereby reducing the communication overhead of the process. These neighborhood sensor nodes sometimes can form clusters which in turn can be arranged in a hierarchical structure. While implementing the TopDisc [39, 114] distributed algorithm there are three methods in which the message passing is done

1. Direct method: When a node receives a topology discovery request, it forwards this message and immediately sends back a response with its neighborhood list along the reverse path.

2. Aggregated method: A node receives a packet, it forwards the request immediately but waits for its children nodes to respond before sending its own response. On receiving responses from its children, it aggregates the data and sends it to its own parent.
3. The third approach is a clustered approach, which forms groups or clusters from the nodes. One node in each cluster is selected as the leader. Only the leader will reply to the topology request. The leader's reply will include the topology information about all the nodes in its cluster.

There are several solutions for topology discovery. The Topology Discovery Algorithm or TopDisc [37] uses of a tree structure, with the root of the tree being the monitoring node, to find the network topology. There are three types of nodes: White (undiscovered), Black (cluster head) and Grey (neighbor of black node).

At first all the nodes remain white except the root. The root starts sending topology request packets to its neighbors. If a white node receives the packet from a black node, it turns grey. If a white node receives the packet from a grey node, then it will wait for a specific time. If it gets another request from a black node within that time, it will turn grey otherwise it will turn black. The black and grey nodes ignore any further request. As the topology request is propagated, neighborhood sets will be generated. This is done by finding the set coverage with a greedy approximation algorithm.

Another topology discovery algorithm is Sensor Topology Retrieval at Multiple Resolutions or STREAM [38]. Using the Wireless Multicast Advantage, STREAM detects the presence of neighboring nodes by eavesdropping on the communication channel. This allows STREAM to create an approximate topology by getting neighborhood lists from a subset of nodes.

Evolving networks of ad-hoc wireless sensing nodes rely heavily on the ability to establish position information. The algorithms presented [145] herein rely on range measurements between pairs of nodes and the a priori coordinates of sparsely located anchor nodes. Clusters of nodes surrounding anchor nodes cooperatively establish confident position estimates through assumptions, checks, and iterative refinements. Once established, these positions are propagated to more distant nodes, allowing the entire network to create an accurate map of itself. Major obstacles include overcoming inaccuracies in range measurements as great as 50%, as well as the development of initial guesses for node locations in clusters with few or no anchor nodes. Solutions to these problems are presented using position error as the primary metric. Algorithms are compared according to position error, scalability, and communication and computational requirements. Early simulations yield average position errors of 5% in the presence of both range and initial position inaccuracies.

For GLIDER [48], gradient landmark-based distributed routing in which a novel naming/addressing scheme and associated routing algorithm is designed for WSN. Here the nodes are fixed (though their geographic locations are not necessarily known), and that each node can communicate wirelessly with some of its geographic neighbors - a common scenario in sensor networks. In GLIDER which has a preprocessing phase that discovers the global topology of the sensor field and, as a byproduct, partitions the nodes into routable tiles - regions where the node placement is sufficiently dense and regular that local greedy methods can work well. Such global topology includes not just connectivity but also higher order topological features, such as the presence of holes. GLIDER addresses each node by the name of the tile containing it and a set of local coordinates derived from connectivity graph distances between the node and certain landmark nodes are associated with its own and neighboring tiles. Here the tile adjacency graph is used for global route planning and the local coordinates for realizing actual inter- and intra-tile routes.

For PEDAMACS [45], the protocol first enables the access point to gather information about the network topology based on the location information. It then calculates and broadcasts a periodic schedule, which determines when each node should listen for incoming packets and when it should transmit its own packets or those received from ‘upstream’ nodes; the rest of the time, the node ‘sleeps’. When a change in network topology is detected, the access point repeats the process of topology discovery and schedule determination.

For wireless ad-hoc sensor networks (WASNs) [106], localized algorithms are used as a special type of distributed algorithms where only a subset of nodes in the WASN participate in sensing, communication, and computation. WASN was developed for generic localized algorithm for solving optimization problems in wireless ad-hoc networks that has five components: (i) data acquisition mechanism, (ii) optimization mechanism, (iii) search expansion rules, (iv) bounding conditions and (v) termination rules. The main idea is to request and process data only locally and only from nodes who are likely to contribute to rapid formation of the final solution. The approach enables two types of optimization: The first, guarantees the fraction of nodes that are contacted while optimizing for solution quality. The second, provides guarantees on solution qualities while minimizing the number of nodes that are contacted and/or amount of communication. The localized optimization approach in WASN is applied to two fundamental problems in sensor networks: location discovery and exposure-based coverage which the algorithm solves successfully.

So the summary of TopDisc algorithm is that it selects a set of distinguished nodes based on location information, and constructs a reachability map based on their information. TopDisc logically organizes the network in the form of tree of clusters which is rooted at the monitoring node. It's seen that is TopDisc algorithm is very efficient for data dissemination and aggregation, duty cycle assignments and network state retrieval. TopDisc is completely distributed, uses only local information and is highly scalable.

2.1.2 Topology Clustering Taxonomy

Clustering algorithms are used to decrease the number of nodes that transmit data to the base station (BS). These algorithms arrange the nodes deployed in the WSN into groups or clusters. One node in each cluster is identified as the leader of the cluster or the cluster head (CH). The nodes that are in a cluster, but are not cluster head, become member nodes of that cluster. The member nodes will transmit their data to their cluster head, which is typically within only a short distance thus consuming less energy. In this branch of taxonomy the emphasis is on the clustering characteristics of the network. The following are listed clustering characteristics for protocol design for the network:

1. Some clustering protocols [161, 114, 169, 46, 113, 33, 100, 73] are used to derive hierarchical networks.
2. Both centralized [107, 73] and distributed [62, 54] computing techniques are used to design the clustering protocol.
3. Some clustering protocols are designed to target specific area for connectivity and coverage issues [52]
4. Some clustering protocols use hybrid schemes like soft computing techniques [52,54], Machine learning approaches [8, 150], and physical analysis using Scale free networks [124, 190, 73], in order to reduce the topology.

Using the above techniques for clustering in wireless sensor networks, they are sometimes classified as [99] Cluster-Construction Based Clustering Routing Protocols and Data-Transmission Based Clustering Routing Protocols or single hop clustering and multi hop clustering [114]. But using the later classification one has better understanding on the mechanics of clustering rather than using the former as used in application.

1) Single-hop Clustering Algorithms

One of the most successful clustering algorithms is LEACH [161]. In LEACH all the nodes die at almost same time. It selects the cluster heads based on the remaining energy in the nodes and also rotates the cluster heads periodically. Thus it guarantees a certain network lifetime while minimizing the energy consumption by the sensor nodes. Heinzelman, et. al. [58] has shown that LEACH “successfully distributes the energy-usage among the nodes in the network such that the nodes die randomly and at essentially the same rate”. Cluster members send the data to its cluster head. The cluster head will fuse all the data and then transmit one message to the base station, containing the data for its cluster.

The disadvantage of LEACH is the cluster heads forward the data to the base station which is in a single hop distance but may be long. Overhead regarding clusters creation is also a drawback. LEACH can also be extended to be hierarchical, so that cluster heads communicate with a higher-level cluster head instead of directly with the base station. M-LEACH [176] is an implementation of LEACH for multi-hop networks, where a node is multiple hops from its cluster head.

Hybrid Energy-Efficient Distributed clustering (HEED) [180], introduced by Younis and Fahmy, is a multi-hop WSN clustering algorithm which brings an energy-efficient clustering routing with explicit consideration of energy. Different from LEACH in the manner of CH election, HEED does not select nodes as CHs randomly. The manner of cluster construction is performed based on the hybrid combination of two parameters. One parameter depends on the node's residual energy, and the other parameter is the intra-cluster communication cost. In HEED, elected CHs have relatively high average residual energy compared to MNs. Additionally, one of the main goals of HEED is to get an even-distributed CHs throughout the networks. Moreover, despite the phenomena that two nodes, within each other's communication range, become CHs together, but the probability of this phenomena is very small in HEED.

The advantages of the HEED protocol are as follows: (1) It is a fully distributed clustering method that benefits from the use of the two important parameters for CH election; (2) Low power levels of clusters promote an increase in spatial reuse while high power levels of clusters are required for inter-cluster communication. This provides uniform CH distribution across the network and load balancing; (3) Communications in a multi-hop fashion between CHs and the BS promote more energy conservation and scalability in contrast with the single-hop fashion, i.e., long-range communications directly from CHs to the sink, in the LEACH protocol [58].

ABCP [60] or Access-Based Clustering Protocol designed the clustering operation from a protocol point of view. It defines the message formats, describes how a node responds when a message arrives, and specifies how a node handles errors. This algorithm is a “simple broadcast request-response with first-come-first-serve selection”. There are many advantages to using ABCP. It does not require any location information. Cluster heads will fuse the data of its member nodes before transmitting the data to the base station that shortens the message to be sent and it is stable even during topology changes, even during the cluster formation process.

Another request-response with first-come-first-serve selection for cluster formation is ABEE [59] or Access-Based Energy Efficient cluster algorithm. This algorithm is very similar to ABCP but is based primarily on location. ABEE will try to balance the residual energy in all the nodes by periodically rotating the role of the cluster head. The new cluster head is selecting by treating the "whole cluster as an entity and each node stands for particles with equal mass to form the entity" .ABEE improves the lifetime of the network when compared to ABCP. According to [59] there is 92.3% lifetime enhancement over the ABEE protocol and around 50% gain in the lifetime of the network coverage".

Energy Efficient Clustering Scheme (EECS) was proposed by Ye et al. [178, 176], is a clustering algorithm which better suits the periodical data gathering applications. EECS is a LEACH-like scheme, where the network is partitioned into several clusters and single-hop communication between the CH and the BS is performed. In EECS, CH candidates compete for the ability to elevate to CH for a given round. This competition involves candidates broadcasting their residual energy to neighboring candidates. If a given node does not find a node with more residual energy, it becomes a CH. Different from LEACH for cluster formation, EECS extends LEACH by dynamic sizing of clusters based on cluster distance from the BS.

The advantages of EECS are summarized as follows: (1) Based on energy and distance, EECS constructs balancing point between intra-cluster energy consumption and inter-cluster communication load; (2) Clustering is performed by dynamic sizing based on cluster distance from the BS. This addresses the problem that clusters with a larger distance to the BS require more energy for transmission than those with a shorter distance, and bring about low message overheads and uniform distribution of CHs compared to LEACH

2) Multi-hop Clustering Algorithms:

PEGASIS [96] or Power-Efficient Gathering in Sensor Information Systems lowers the overhead of cluster formation in LEACH. The key idea of PEGASIS it to form a chain among the nodes and take turns transmitting the data to the base station. This allows each node to communication only with a closest neighbor, thus consuming less energy.

There are several assumptions in the PEGASIS algorithm. First, it assumes that all nodes have global knowledge of the network. This allows them to create the best chain using the greedy algorithm and each node will know its neighbor nodes. It also assumes that all nodes employ the greedy algorithm and that the radio channel is symmetric.

The advantage of PEGASIS is since nodes only receive and transmit to its neighbors, and they form a chain, each node will only transmit and receive one packet of data in each round. If a node fails the chain can be reconstructed with the remaining nodes. This makes PEGASIS robust to node failures.

Another clustering algorithm is the Energy Efficient Clustering Scheme or EECS [178]. The goals were to create a fully-distributed, load-balancing clustering algorithm that had little overhead. It is very much like LEACH but can better balance the load among the clusters and cluster heads. In EECS, there is only one cluster head within a certain range with a high probability. According to [178], the control overhead across the network is O(n). This chapter also indicated that EECS will prolong the network lifetime over 35% when compared to LEACH. The energy utilization rate is also better in EECS because "EECS always achieves the well distributed cluster heads while considering the residual factor; further, we consider to balance the load among the cluster heads with weighted function".

Base-Station Controlled Dynamic Clustering Protocol (BCDCP), introduced by Muruganathan et al. [107], is a centralized clustering routing protocol with the BS being capable of complex computation. The main idea of BCDCP is the cluster formation where each CH serves an almost equal number of MNs to balance CH overload and uniform CH placement throughout the network.

At the beginning of cluster setup, the BS receives information on the residual energy from all the nodes in the network. Based on this information, the BS first computes the average energy level of all the nodes in the network, and then chooses a set of nodes whose energy levels are above the average value. Only the nodes from the chosen set, i.e., those with sufficient energy, can be elected CHs for the current round, while those with low energy can prolong their lifetime by performing the task of ONs. Based on the chosen set, the BS computes the number of clusters and performs the task of clustering, which is accomplished in terms of an iterative cluster splitting algorithm. This algorithm first splits the network into two sub-clusters, and proceeds further by splitting the sub-clusters into smaller clusters. This process will be repeated until the desired number of clusters is achieved. At each iteration of cluster splitting, two nodes that have the maximum separation distance are chosen for CHs from the chosen set where all the nodes are eligible to become CHs. Then, each of the remaining nodes in the current cluster is grouped with one CH or the other, whichever is closest. After balancing the two groups which have approximately the same number of nodes, the two sub-clusters are formed.

In BCDCP, a multi-hop routing scheme is adopted to transfer the sensed data to the BS. Once the clusters and the CHs have been identified, the BS chooses the lowest-energy routing path and transfer information to the nodes along with the details on cluster groupings and selected CHs. The routing paths are selected by first connecting all the CHs by means of the Minimum Spanning Tree (MST) approach , which minimizes the energy consumption for each CH, and then randomly choosing one CH to forward the data to the BS.

2.1.3 Topology Sleep Cycle Taxonomy

To conserve energy, in a node is to only have it powered on when necessary; the node would be powered off or put to sleep all other times.

The following are the techniques in this category that are based on the following conditions

1. Synchronized vs. Non-synchronized DC: If global or local synchronization is assumed for a DC-WSN, the model is amenable to graph-theoretical characterizations because one may augment the original connectivity graph of the WSN by associating a binary (active or dormant) state with each node. As a result, most combinatorial solutions make such an assumption [173, 175, 138, 68, 165, 88, 77]. If one wants to eliminate the overhead for synchronization, the working periods of different nodes may not align with each other [70]. Therefore, the residual active/dormant time is a random variable, and hence a stochastic modeling technique has to be used [79, 69].

2. Generalized vs. Simplified DC: Though a duty-cycling model often assumes an identical and fixed working period T, each node in general can determine its active/dormant schedule without any constraints. Such a generalized DC model is frequently adopted, as indicated in the literature [173], [138], [77].However, simplified DC model is also considered to facilitate algorithm design; such simplifications often impose certain restrictions on the active/dormant schedules within a working period. For example, a single-active-time-slot model is used in [70], [68], [165], i.e., there exists only one active time slot in a working period of any node. Both [88] and [170] allow for multiple active time slots but [88] requires the (variable number of) active time slots to be consecutive in a working period while [137] assumes that the proportion of active time slots in a working period of any node equals to a predefined constant. Following the convention of stochastic analysis, the work in [137] assumes that each node wakes up independently according to the Poisson process.
3. Static vs. Dynamic DC: Many algorithms we discuss in this here assume that the active/dormant time slots of any network node are static, i.e., the active/dormant schedule is pre-determined for each node and cannot be changed by the algorithms [173], [70], [137], [138]. However, there also exist algorithms adopting dynamic duty cycling models, such as [151],[175], [137], [157]. The common idea behind these proposals is that the active/dormant schedules of sensor nodes can be dynamically controlled such that the nodes are awaken only when they are needed, hence more energy can be conserved because the power consumption for idle-listening is reduced and retransmissions caused by collisions are limited. Besides energy conservation, other optimization goals affected by dynamic DC (such as latency and capacity) are also considered in these proposals.
4. Reliable vs. Unreliable Links: As with conventional WSNs, link reliability issue persists in DC-WSNs. However, considering both DC and link reliability issue may significantly complicates the problem. Therefore, many research proposals neglect the latter, with some exceptions [173, 175], [151,174 184]. In particular, most combinatorial approaches tends to avoid the link reliability issue [68, 165, 88, 77], as it simply adds the dimension of the resulting problems.

Using the above formulations some of the well-known protocols have been devised along this directions such as

Asynchronous Duty-cycle Broadcasting (ADB) [149] which uses the asynchronous duty cycling in wireless sensor network in MAC protocols, since it can greatly reduce energy consumption and requires no clock synchronization. However, existing systems using asynchronous duty cycling do not efficiently support broadcast-based communication that may be used, for example, in route discovery or in network-wide queries or information dissemination. ADB differs from traditional multihop broadcast protocols that operate above the MAC layer, in that it is integrated with the MAC layer to exploit information only available at this layer. Rather than treating the data transmission from a node to all of its neighbors as the basic unit of progress for the multihop broadcast, ADB dynamically optimizes the broadcast at the level of transmission to each individual neighbor of a node, as the neighbors asynchronously wakeup.

Opportunistic Flooding in Low Duty Cycle with Unreliable Links (OFLDCUL) [D29], which flooding service has been investigated extensively in wireless networks to efficiently disseminate network-wide commands, configurations, and code binaries. However, little work has been done on low-duty-cycle wireless sensor networks in which nodes stay asleep most of the time and wake up asynchronously. In this type of network, a broadcasting packet is rarely received by multiple nodes simultaneously, a unique constraining feature that makes existing solutions unsuitable. Starting with an energy-optimal tree structure, probabilistic forwarding decisions are made at each sender based on the delay distribution of next-hop receivers. Only opportunistically early packets are forwarded via links outside the tree to reduce the flooding delay and redundancy in transmission. A forwarder selection method to alleviate the hidden terminal problem and a link-quality-based back off method to resolve simultaneous forwarding operations. Compared with Improved Traditional Flooding, the design achieves significantly shorter flooding delay while consuming only 20-60% of the transmission energy.

Flooding in low-duty-cycle wireless sensor networks is very costly due to asynchronous schedules of sensor nodes. To adapt existing flooding-tree-based designs for low-duty-cycle networks, we shall schedule nodes of common parents wake up simultaneously. Traditionally, energy optimality in a designated flooding-tree is achieved by selecting parents with the highest link quality. In this work, we demonstrate that surprisingly more energy can be saved by considering link correlation.

. A novel flooding scheme, named Correlated Flooding [57], is then designed so that nodes with high correlation are assigned to a common sender and their receptions of a broadcasting packet are only acknowledged by a single ACK. This unique feature effectively ameliorates the ACK implosion problem, saving energy on both data packets and ACKs. It is seen that Correlated Flooding saves more than 66% energy on ACKs and 15%~50% energy on data packets for most network settings, while having similar performance on flooding delay and reliability.

As all the approaches described above require the system energy to find the neighboring nodes and then aggregating or clustering which is one of the greatest drawback logically and physically for implementing static wireless sensor network. In order to improve upon, the authors in [146] proposed an intelligent method based on Self Organizing Map neural networks that optimize the routing in the terms of energy conservation and computation power of each node. This algorithm has been designed for a wireless sensor node called MODABER. The assumption is that every node has an importance due to its role in routing so that the nodes which are used more than other nodes in routing have more importance due to their positions. They defined a Network Life Time (NLT) parameter which is sum of the nodes importance in routing at time t and the amount of energy consumption of node for routing. They used a self-organizing (competitive) neural network to decide for every node containing the data packet and participate in routing or dropping the packet. The Self Organizing Map (SOM) learning algorithm is used for training of neural network. As soon as a packet arrives, its feature vector will be extracted and this vector is sent to self-organizing NN of that node as input. The goal is to maximize NLT parameter. After winning of node in competition against other nodes, it is allowed to send the packet and participate in routing. Otherwise it should drop the packet. Since the learning algorithms of SOM's generally obey from linear computations, they believe that this method can be efficient to wireless nodes due to their limited computation and energy powers. While implementing SIR [15], SOM neural network is introduced in every node to manage the routes that data have to follow. Here they have implemented SOM using QoS metrics viz. latency, throughput, error-rate and duty-cycle related to each node as input samples with an output layer neuron. The samples allocated in the SOM form groups, in such a way that all the samples in a group have similar characteristics (latency, throughput, error-rate, and duty-cycle). This approach is used to obtain clusters with specific QoS values, lower value of QoS should be avoided as it depicts the worst case scenario.

All the above two schemes have shown their respective success but the parameters or metrics assumed are not difficult but time consuming to evaluate. This situation adds up to the cost of computation.

So in this regard we [118] have designed an approach using Self Organizing Map (SOM) which makes the necessary high cost computation offline and only the implementation online. The scheme details can be found below.

This approach is selection by Self Organizing Map, in which a continuous input space of activation patterns (the spatial co-ordinates of sensors) is mapped onto a discrete output space of neurons (the would be selected spatial co-ordinate of sensors) by the process of competition among the neurons in the network. At the same time if the coordinate obtained do not map to the original ones, then by using k-nearest-neighbor algorithm the mapping is done as well we can iteratively remove the redundancy from the list of spatial coordinates.

Important advantages of this scheme is in addition to low power consumption include simplicity, inherent robustness to node or link failure and changing network geometry (in case of battery depletion), reduced redundant packet transmissions and implicit network reconfiguration. The only disadvantage is the need for sufficient density to maintain network operation.

The primary use of energy in WSN’s is the transmission of data. Another way to conserve energy is to have fewer nodes transmit data to the base station, which is the device collecting the application data. Clustering algorithms are used to decrease the number of nodes that transmit data to the base station (BS). These algorithms arrange the nodes deployed in the WSN into groups or clusters. One node in each cluster is identified as the leader of the cluster or the cluster head (CH). The nodes which are in a cluster, but are not cluster head, become member node of that cluster. The member nodes will transmit their data to their cluster head, which is typically within only a short distance thus consuming less energy. The cluster head will then forward the data received from each of its member nodes to the base station. Only the cluster heads will transmit data to the base station. Many clustering algorithms will also aggregate or fuse the data received from the member nodes at the cluster head resulting in less data being transmitted from each cluster head to the base station. As less data is transmitted, less energy is used.

So in order to minimize the energy consumption authors [31, 154] have proved SOM to be an effective platform for visualization of high dimensional data and hence SOM as the first level of abstraction in clustering has some clear advantages. First the original data set is represented using smaller set of prototype vectors, which allows efficient use of clustering algorithms to divide the prototypes into groups. Secondly, reduction of the computational cost/transmission power is especially important for hierarchical algorithms allowing clusters of arbitrary size and shape.

In another approach the authors [30] have analytically used the theory of complex network to quantify some of the observed properties of topology control algorithms. To build this framework, probabilistic approach of Li-Chen model [167] of Local world model is used to mimic the wireless sensor network. And the dynamics of clustering was addressed by the concept of preferential and anti-preferential attachment. The anti-preferential removal mechanism is more reasonable for deleting links that are anti-parallel with the preferential connection [30, 167, 2]. It is also consistent with the functioning of clustering algorithms that runs in rounds in wireless sensor networks. The wireless nodes that do not have enough energy, that is, the dead nodes, are to be removed from the system. Thus, anti-preferential [30] removal phenomenon is reasonable for clustering algorithms. Finally combining the mathematical realizations of the above mentioned facts in mean field theory, we obtain the distribution function as the degree distribution P(k) , where P(k)is the probability of the node has k edges. This distribution is further minimized with respect to the anti-preferential attachment , which during the evolution process tends to zero as this phenomenon of non-attachment to a preferential neighbor is absent for wireless sensor networks. This consideration reduces the distribution function to yield.

$$p = 0.5 * \left\lfloor \sqrt{\frac{k+3}{k-1}} - 1 \right\rfloor \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

The above expression is called the probability of clustering in the network.

Now using the above expression in optimizing matrix [30] the NLT of one of the most widely used clustering protocol LEACH and LEACH-C have increased by 300% and 150% respectively, which is best by any standards over any protocol built over LEACH.

As it can be seen through the above discussion that energy conservation is critical in all WSN's, but may be more critical in a long-term deployment. If the intended lifetime of the WSN is relatively short, then an algorithm that conserves less energy but sacrifices less latency may be appropriate. This characteristics may be understood through the discussions pertaining to the next sections which deals with the topology of perseverance.

2.2 Techniques of Topology Perseverance:

There are three topology perseverance techniques

1. Static
2. Dynamic
3. Hybrid

Static Topology Perseverance

In static topology perseverance technique all possible topologies are calculated and stored. The topology just switched from one to another when needed.

Advantage: As all the topologies are pre-calculated, so the transition among the available topologies is fast. It also saves the overhead of topology construction every time the switching is happened.

Disadvantage: But it has some disadvantages also, it cannot be known in advance that how the nodes will lose their energy. Due to extensive use of some nodes in one topology makes them unavailable for the next. It also takes some more time at the beginning to calculate all the possible topologies.

- Dynamic Topology Perseverance

Dynamic topology perseverance technique creates a new topology when necessary on the fly.

Advantage: It has the current information about the network that helps it to make an appropriate reduced topology.

Disadvantage: It takes more resource and time every time it runs.

- Hybrid Topology Perseverance

It uses both static and dynamic techniques. It calculates all the possible reduced topologies at the beginning that is during the first topology construction phase (static approach), but if it cannot implement it because of the node failure or connectivity failure, it creates a new topology on the fly (dynamic approach).

It inherits the advantages and disadvantages of both the techniques.

Design Issues:

The following are important considerations for effective topology perseverance mechanisms:

- Distributed: Being distributed rather than central, the algorithms can save more energy because in central approach some nodes may have to communicate long distances, whereas in distributed approach there are more sinks and base stations resulted in communications with only the closest ones. So there will be even distribution of energy among the sensor nodes.
- Local information: Nodes should be able to make topology control decisions locally. This reduces the energy costs and makes the mechanism scalable.
- Need of location information: The need of extra hardware or support mechanisms adds to the cost in terms of dollars and energy consumption. One example is the need of location information, which might be provided by GPS devices or localization protocols.
- Robust to node failures and node mobility: The algorithm will be more successful if it is robust to node failures. Sensor nodes are often prone to failure due to running out of energy, hardware failures or simply the node being destroyed due to harsh

conditions. Sometimes these sensor nodes move by nature of application or by accidentally. The protocol or algorithm should be developed so it is robust to node mobility and node failures.

- Low overhead: Topology management mechanisms must work with very low message overhead, so they are energy-efficient and can be run many times as part of the topology maintenance cycle.
- Low Complexity: Topology management algorithms must have a low computational complexity, so they can be run in wireless sensor devices.
- Low Convergence Time: During the topology perseverance process a current topology will be replaced by a new one, therefore there will be a transition time during which the network might not be active. This time must be as small as possible. Static techniques offer a clear advantage in this aspect, as the new topology has already been calculated. In dynamic techniques, this time will be longer and depends on the convergence time of the topology construction mechanism.
- Memory Consumption: The memory of wireless sensors devices is limited. The topology maintenance static techniques need to have a considerable amount of memory to store all the pre-calculated topologies.

2.2.1 Triggering Criteria

The topology perseverance mechanism may be static, dynamic, or hybrid, global or local, there is one important question related to all: what is the criterion or criteria that will be used to trigger the process of changing the current topology? The triggering criteria, which may have important implications in terms of energy savings as well as coverage, reliability, and other important metrics, may be based on one of the following choices:

- Time based: In time-based topology perseverance, the current topology is changed every time a timer expires. The amount of time is usually fixed and pre-defined. This is a very critical variable. It can’t be too short or too long. As being too short the switching of the topology perseverance algorithms will be very often, resulted in a waste of energy of the sensor nodes. On the other hand too long a time can make some important nodes unavailable.

- Energy based: Sometimes we can use the remaining energy of the nodes as the triggering criterion of the topology perseverance techniques. There should be a threshold value, on reaching it the topology will be triggered to change. Again this value is critical too for same reasons explained before.
- Random based: In random based topology perseverance a random variable is used to switch the current topology.
- Failure based: The failure based technique triggers the topology change only if one or some of the wireless sensor nodes failed. But failure detection and notification technique should be there.
- Density based: In density based triggering criterion the node degree of the nodes can be an important metric.
- Combinations: The criteria can be used in combination as well. Such as we can use energy and time or energy and failure to change the wireless network topology

The following were the issues pertaining to perseverance issues of wireless sensor network which makes it possible to increase the NLT of the network with seamless connectivity and coverage.

2.2.2 Examples of Topology Perseverance Methods

Span [D31], a power saving technique for multi-hop ad hoc wireless networks that reduces energy consumption without significantly diminishing the capacity or connectivity of the network. Span builds on the observation that when a region of a shared channel wireless network has a sufficient density of nodes, only a small number of them need be on at any time to forward traffic for active connections. Span is a distributed, randomized algorithm where nodes make local decisions on whether to sleep, or to join a forwarding backbone as a coordinator. Each node bases its decision on an estimate of how many of its neighbors will benefit from it being awake, and the amount of energy available to it. We give a randomized algorithm where coordinators rotate with time, demonstrating how localized node decisions lead to a connected, capacity-preserving global topology. Improvement in system lifetime due to Span increases as the ratio of idle-to-sleep energy consumption increases. Simulations show that with a practical energy model, system lifetime of an 802.11 network in power saving mode with Span is a factor of two better than without. Additionally, Span also improves communication latency and capacity.

T-Man [75] a gossiping based framework for topology management which is mainly used for constructing and maintaining a large class of topologies. In this framework, a topology is defined with the help of a ranking function. The nodes participating in the protocol can use this ranking function to order any set of other nodes according to preference for choosing them as a neighbor. This simple abstraction makes it possible to control the self-organization process of topologies in a straightforward, intuitive and flexible manner. At the same time, the T-Man protocol involves only local communication to increase the quality of the current set of neighbors of each node. On the similar approach [116] a probability based gossiping framework is created which helps in reducing the control messages in the network and, thereby reducing the energy consumption in the network.

Forcing all IP packets to carry correct source addresses can greatly help network security, attack tracing, and network problem debugging. However, due to asymmetries in today's Internet routing, routers do not have readily available information to verify the correctness of the source address for each incoming packet. A new protocol, named SAVE [92], which can provide routers with the information needed for source address validation. SAVE messages propagate valid source address information from the source location to all destinations, allowing each router along the way to build an incoming table that associates each incoming interface of the router with a set of valid source address blocks.

Another important aspect of WSN is the deployment of sensor networks which is concerned with setting up an operational wireless sensor network in a real-world setting. Unfortunately, deployment is a labor-intensive and cumbersome task as environmental influences often degrade performance or trigger bugs in the sensor network that could not be observed during lab tests.

Chapter 3

Energy Efficient Topology Management using ML in WSN

Preserving the energy or battery power of a wireless sensor network is of major concern. In such types of network, the sensors are deployed in an ad hoc manner, rather than in any deterministic way. This chapter is concerned with applying standard routing protocols into wireless sensor networks by using neural network modified topology, which proves to be energy efficient as compared with unmodified topology. Neural networks have been proved to be a powerful tool in the distributed environment. Here, to capture the true distributed nature of the Wireless Sensor Network (WSN), the neural network's Self-Organizing Feature Map (SOFM) is used.

3.1. Introduction

Sensors in the Wireless Sensor Networks mainly use batteries. Very often these batteries are non-rechargeable or non-replaceable due to the geographical location of those sensors. Therefore, energy preservation of sensor nodes is a crucial issue in reducing the quick exhaustion of the energy of sensor nodes and thereby in prolonging the overall network lifetime. The limited resources of the sensor nodes need to be spent judicially so that the minimum energy for this energy-consuming task is required. Several techniques have been proposed so far, which emphasize the energy-efficient routing protocol. Most of them tend to utilize the same single optimal path for each communication time [63, 24]. A straight line routing protocol has been mentioned in [22] which achieves routing in WSN without broadcasting. But using only a single path is prone to node and link failures because of the depletion of batteries. In case of failure due to any reason, an alternative route has to be discovered for maintaining the continuous transmission from source to destination. This technique requires extra energy for the route discovery process. Using multiple paths in the wireless sensor network can enhance the overall efficiency, reliability, and integrity of the network. It can prove to be an effective way of distributing the traffic load evenly over the network. Most of the multiple paths routing protocols to date are based on the classic on-demand single path routing methods [140, 103] such as AODV and DSR. They have their own technique of selecting multiple routes. Some papers are concerned with the node energy in the construction of multiple paths [85, 1].

But all the aforementioned approaches are most suitable for static topologies and have some problems in common. They flood the route request to the network at the same time over the whole network, thus increasing the overhead and so the probability of route congestion becomes high. They also waste the sensor node’s energy unnecessarily. Moreover, when several alternative paths transmit data packets arbitrarily, there exists a probability of high packet loss rate even if node-disjoint multi paths are used, resulting in a suboptimal Computation-Communication trade-off. For dynamic topology an efficient distribution method to form a weighted connected dominating set (the backbone) could be achieved through constant approximation ratio on cost optimization [181]. Whereas some researchers believe the shape of the topology is as important as topology control for 3D sensor networks, it has been proved that some of the structures could guarantee the power efficiency of all paths.

Any kind of time development (be it deterministic or essentially probabilistic) which can be analysed in terms of probability deserves the name of stochastic process. This process offers a simple, robust, and ultra-low-power solution for many sensor network applications. A “stochastic” sensor network is proposed in [84] in which a sensor node operates normally and consumes stored energy in the wake mode until the energy is depleted, and then ceases processing and reverts back to the sleep mode while scavenging the environment for usable sources of energy. When the recharging process is complete, the node resumes normal operation. This behaviour of having asynchronized wake-sleep modes among sensor nodes constitutes a stochastic sensor network. One of the biggest problems with stochastic sensor networks is redundant packet transmissions to sustain network traffic via stochastic flooding [139].

Keeping the degree of complexity of the above-mentioned approach in mind, the concept of SOFM from a neural network is taken. The unique property of SOFM is the mapping of continuous input space of some certain distribution functions to a discrete output space. This discrete output space consists of the modified topology of the network that will be used for information dissemination [35].

The rest of this chapter is organized as follows; we briefly discuss a Self-Organizing Feature Map (SOFM) network model and assumptions in Section 2 and an algorithm formulation in Section 3. Here we also have a target system which will be discussed in Section 4 and have discussed implementation details through proper interpretation of the output graphs from PROWLER in Section 5. Finally, we conclude in Section 6, pointing out future research directions.

3.2. Related Reviews:

3.2.1. Flooding Protocol

The first category of routing protocols are the multi-hop flat routing protocols. In static flat networks, each node plays the same role and sensor nodes help in the sensing activities. Due to the large number of such nodes, it is not feasible to assign an identifier with respect to position and energy. This consideration has led to data-centric routing, where the sink or the base station sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is being requested through queries, attribute-based naming is necessary to specify the properties of data.

3.2.2. Spanning Tree Protocol

Spanning Tree Protocols are the most common routing trees in the networks and particularly in WSNs, because they represent the routing structure in terms of time and energy. These time and energy representations are efficient in a way. There are many algorithms used to construct Spanning Tree Protocols, including the algorithms of Bellman–Ford and Dijkstra. Moreover, WSNs in general do not have centralized management and fixed infrastructure. The authors Bertsekas and Gallager proposed a distributed asynchronous version of the Bellman–Ford algorithm for distribution systems such as wireless sensor networks. This variant of Bellman–Ford (BF) presents fast convergence, in the absence of any synchronization overhead, and easy adaptation to topological network changes, but it is not efficient for networks composed of a large number of nodes or dense networks, because of the excessive number of messages required for the tree construction.

3.2.3. K-Nearest Neighbour Algorithm (KNN)[15]

The K-Nearest Neighbour (KNN) algorithm is used to perform the classification. This decision rule provides a simple nonparametric procedure for the assignment of a class label to the input pattern based on the class labels represented by the closest (say, for example, in the Euclidean sense) neighbours of the vector. K-Nearest Neighbour is a classification (or regression) algorithm which, in order to determine the classification of a point, combines the classification of the K nearest points. It is supervised because the attempt is made to classify a point based on the known classification of other points.

It is also a lazy algorithm. What this means is that it does not use the training data points to do any *generalization*. In other words, there is *no explicit training phase,* or it is very minimal. This means the training phase is relatively rapid. Lack of generalization means that KNN keeps all the training data. More exactly, all the training data is needed during the testing phase (this is arguably an exaggeration, but not far from the truth). This is in contrast to other techniques such as SVM, in which all non-support vectors can be discarded without any problem. Most of the lazy algorithms – especially KNN – make decisions based on the entire training data set (or, in the best-case scenario, a subset thereof).

Here are the steps to follow the KNN algorithm

1. Determine parameters K=numbers of nearest neighbours.

2. Calculate the distance between the query instance and all the training samples.

3. Determine the distance and the nearest neighbours based on kth minimum distance.

4. Gather the category of nearest neighbours.

5. Use the simple majority of the category of nearest neighbours as the prediction value of the query instance

3.2.4. Approximate Nearest Neighbour Algorithm (ANN) [11]

The approximate nearest neighbour problem has been considered by Bern [1993]. He proposed a data structure based on quad trees, which uses linear space and provides logarithmic query time. However, the approximation error factor for his algorithm is a fixed function of the dimension.

Computing exact nearest neighbours in dimensions much higher than 8 seems to be a very difficult task. Few methods seem to be significantly better than a brute-force computation of all distances. However, it has been shown that by computing nearest neighbours approximately, it is possible to achieve significantly faster running times (on the order of 10's to 100's) often with relatively small actual errors. ANN allows the user to specify a maximum approximation error bound, thus allowing the user to control the trade-off between accuracy and running time.

The search algorithm begins with a point p selected by choosing a point from a bucket of a k-d tree that contains the query point. We maintain a set of candidates to the nearest neighbour (maintained using a heap) initially containing p. We select the nearest of the candidates that has not already been visited. The algorithm is outlined below:

```
function NN 2(p, q) {
    C := {p};
    nn := p;
    while (C 6= ∅ and termination condition
    not yet met) {
        p := the point of C minimizing dist(q, p);
        C := C − {p};
        for each undiscovered r in N[p] {
            Mark r discovered;
            C := C + {r};
            if (dist(q, r) < dist(q, nn)) nn := r;
        }
    }
    return(nn);
```

}

3.3. SOFM Network Model and Assumptions

The network under consideration consists of *N* number of sensors nodes scattered over 2-dimensional space. For network analysis with Kohonen's Self-Organizing Map, let us assume that the Kohonen's layer consists of *N* neurons. Here we represent a neuron of the neural network to a sensor node of the wireless sensor network (WSN). In this context, we will use the terms neurons and sensor nodes to describe the same thing. Furthermore, we assume the two-dimensional lattice of Kohonen's map represents the area of a wireless sensor network. Input spatial data of the 2-dimensional space as described is assumed to follow Poisson's distribution function for obvious reasons [139]. From the concept of Kohonen's Self-Organizing Feature Maps (SOFM) these input spatial data act as input vectors that learn to classify according to how they are grouped in the input space. They differ from competitive layers in that neighbouring neurons in the self-organizing map learn to recognize neighbouring sections of the input space. Thus, self-organizing maps learn both the distribution (as do competitive layers) and topology of the input vectors on which they are trained.

Learning in a Self-Organizing Feature Map (SOFM) occurs for one vector at a time, independent of whether the network is trained directly or whether it is trained adaptively.

First, the network identifies the winning neuron. Then the weights of the winning neuron, and the other neurons in its neighbourhood, are moved closer to the input vector at each learning step. The winning neuron's weights are altered in proportion to the learning rate. The learning rate and the neighbourhood distance are used to determine which neurons in the winning neuron's neighbourhood are altered during training.

Thus, through Kohonen's learning the winning neuron will be selected which will be treated as a speaker node for a region.

Now this speaker node of the described network has a specific coverage region. All the neighbour nodes reside at a certain distance from a speaker node. It is found during the cooperative process of SOFM that the topological neighbourhood function satisfies the requirement of Gaussian function [135]. So, it may be concluded that the speaker node is surrounded by those nodes which fall in the Gaussian range as decided by previous competitive processes. Lastly, the synaptic adoption process enables the exited neighbouring nodes to increase their individual values of the discriminant function in relation to the input pattern through suitable adjustments applied to their synaptic weights. The adjustments made are such that the response of the speaker nodes to the similar input pattern is enhanced.

Mathematically, it can be simplified as follows:

Step 1. Initialization. Choose random values for the initial weights $w\, j\ (0)$.

Step 2. Finding the Speaker. Find the winning unit j^* by using the minimum-distance Euclidean criterion $j^* = \arg\min \|x\, j(t) - w\, j\|\ ,\ j = 1, \ldots, N$, where $x\, j(t)$ represents the input pattern, N is the total number of units, and · indicates the Euclidean norm.

Step 3. Weights Updating. Adjust the weights of the winner and its neighbours, using the following rule: $w\, j(t + 1) = w\, j(t) + \alpha N\, j^*(t)(x\, j(t) - w\, j(t))$, where α is a positive constant and $N\, j^*(t)$ is the topological neighbourhood function of the winner unit j^* at time t. The neighbourhood function is traditionally implemented as a Gaussian (bell-shaped) function:

$$Nj^*(t) = \left(\frac{1}{\pi\sigma\sqrt{2}}\right)^* \exp\{-(j^* - j)/2\sigma^2\}$$

with σ a parameter indicating the width of the function, and thus the radius in which the neighbours of the winning unit are allowed to update their prototype vectors significantly. It should be emphasized that the success of the map formation is critically dependent on how the values of the main parameters (i.e., α and $N\, j^*(t)$), initial values of weight vectors, and the number of iterations are pre-specified. The Kohonen SOM mainly has implementations based on a single-processor, centralized method.

The adjustments of the speaker nodes produce results; that is, spatial coordinates, which are not amongst any of the input coordinates. So, in order to map the speaker coordinates with respect to input data we use the nearest-*k* neighbour algorithm. The value of *k* for the nearest neighbour algorithm is incremented by one iteratively until the redundancies in the list of participating nodes are removed.

3.4. SOFM Topology Building (SOFMTB) Algorithm

3.4.1. Proposed SOFM Topology Building (SOFMTB) Algorithm.

```
Algorithm SOFMTB (Input Vector)
//Input Vector file contains spatial
//coordinates for
 //sensor distribution on the field.
{
Load Invec: = Input Vector;
SOFM:= Create SOM(n, m);
//Define a SOM whose input data points vary
// from -n to n,  with m nodes.
SOFM:= Train(SOFM, Invec);
//Train the SOM with value from Invec.
FOR i := 1 to m
Save new set points post training in array x;
 Set Actual data := Invec;
Set Ideal data := x;
WHILE (duplicate coordinate)
   t: = KNearestNeighbor/ApproximateNeighbor (Actual data, Ideal data);
 //Ideal data points are mapped with input iteratively
//by means of KNN/ANN until the duplicities are removed.
topology:= t;
Set Node IDs from 1 to m;
Present the topology to link layer protocol
 such as Spanning Tree;
```

ALGORITHM 1

The following are the steps of the SOFM topology building algorithm:

Step 1. Spatial coordinates for sensor distribution on the field are taken as input vectors.

Step 2. SOFM algorithm is used to train the spatial coordinates with the number of neurons equal to the desired number of speaker nodes at the output.

Step 3. An array of new set points in the spatial dimension is returned.

Step 4. The returned array is mapped for real spatial coordinates with the help of the *K*-nearest neighbour algorithm with respect to speaker nodes.

Step 5. The output contains duplicated spatial coordinates. To remove this duplication, we iteratively run the *K*-Nearest-Neighbour algorithm (KNN) or Approximate Nearest Neighbour (ANN) for subsequent values of *K* until the duplication is removed.

Step 6. An array of non-duplicated set points in the spatial dimension is returned.

Step 7. This topology is presented to the link layer protocol, such as a Spanning Tree Protocol.

3.5. Target System

A very successful, low-cost prototype field-node (mote) family was developed at Berkeley. The used variant (MICA) of the Berkeley motes includes an 8-bit, 4 MHz Atmel ATMEGA103 microcontroller, 128 kB program memory, 4 KB RAM, and an RFM TR1000 radio chip capable of providing a 50 kbit/s transmission rate at 916.5 MHz. The motes can also accommodate a set of interchangeable sensors (temperature, light, magneto, sound, etc.) [19].

The motes use a small operating system called TinyOS, designed to provide the necessary services despite very limited hardware resources. It contains a complete network stack with bit-level error correction, medium access layer, network messaging layer, and timing.

The Medium Access Control layer uses a simple Carrier Sense Multiple Access protocol. This waits for a random duration before trying to transmit a packet and then waits for a random backoff interval if the channel is found to be busy. It keeps trying until the transmission can be performed. This simple approach is not as effective as the more sophisticated protocols (e.g., IEEE 802.11, [19, 51]) in terms of collision avoidance, but it certainly consumes less energy and the communication overhead is much smaller.

3.6. Implementation Details

3.6.1 Illustration of the Algorithm:

The Dataset Input_data1.txt contains a random set of 50 points in space without any defined topology. The execution of the algorithm will lead to defining a topology that can be viewed through various steps as depicted in the figures below.

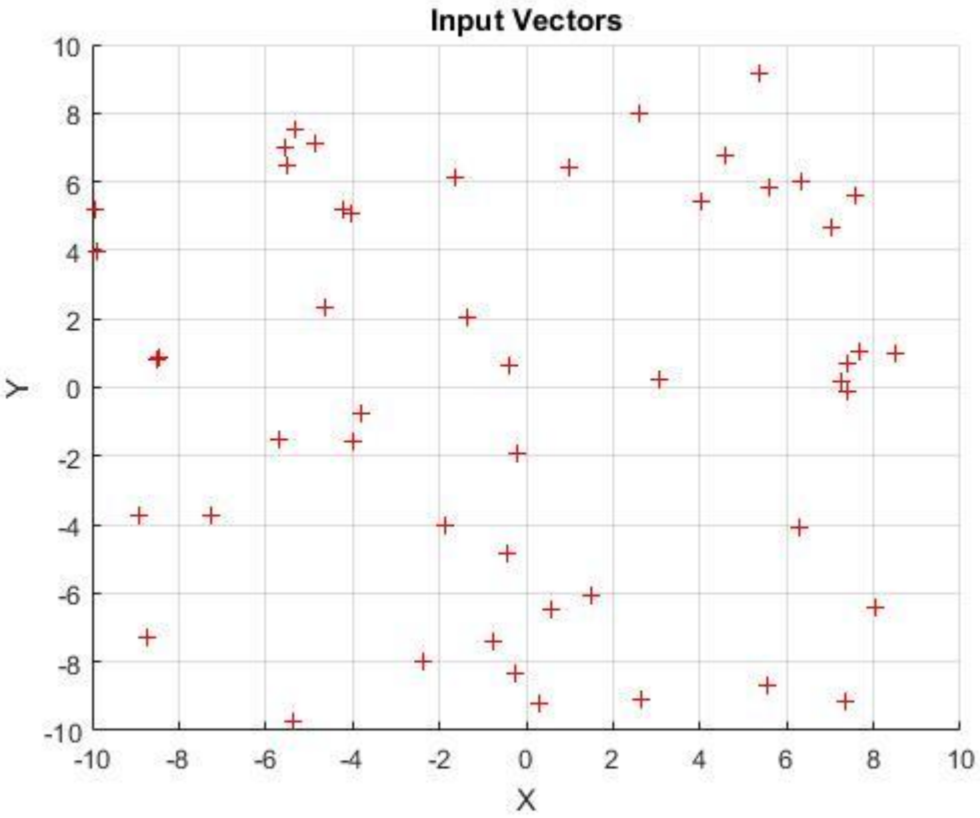


Figure 3.1 Plot of the Data Set Input_data1.txt in an X-Y Plane

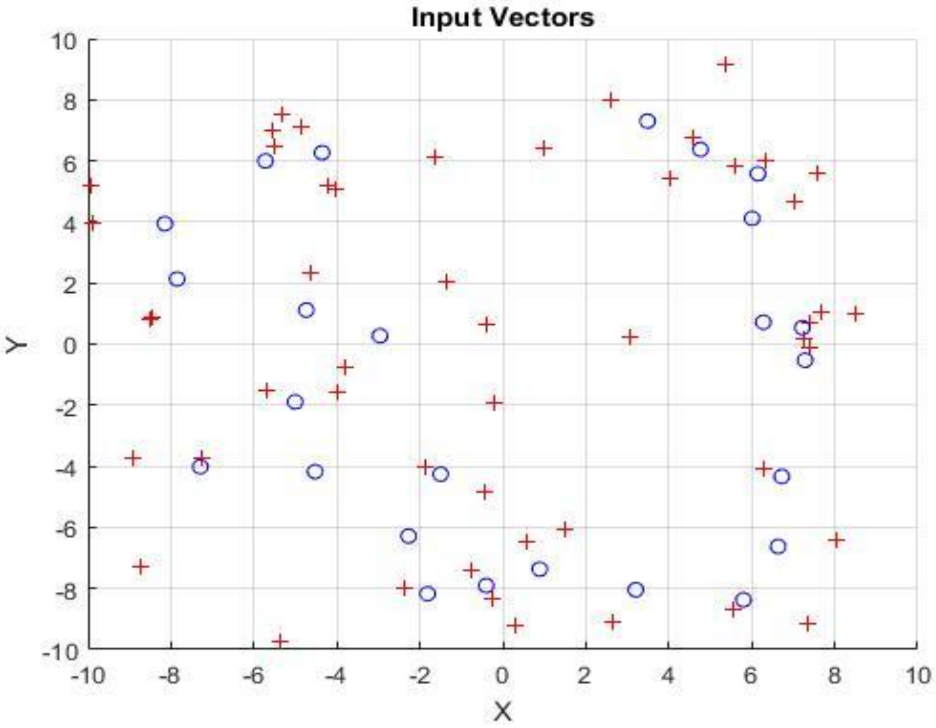


Figure 3.2 Plot of the SOFM generated points (in blue circle) and Data Set Input_data1.txt (red cross) in an X-Y Plane

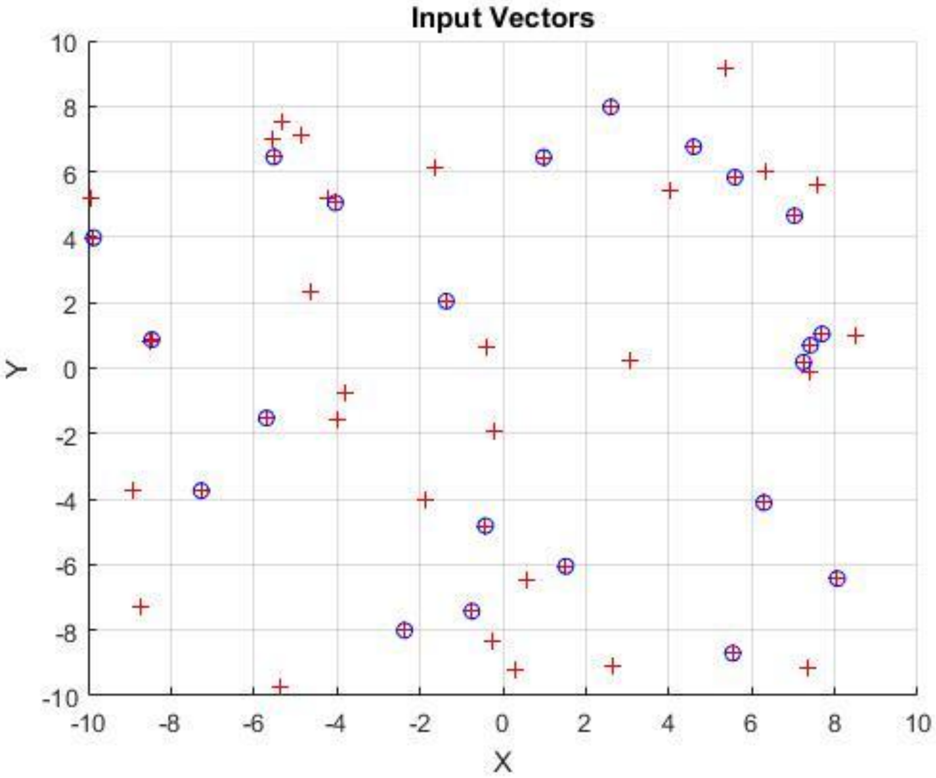


Figure 3.3 Plot of the matched SOFM generated points (in blue circle) and Data Set Input_data1.txt (Red Cross) in an X-Y Plane using k-Nearest Neighbour algorithm.

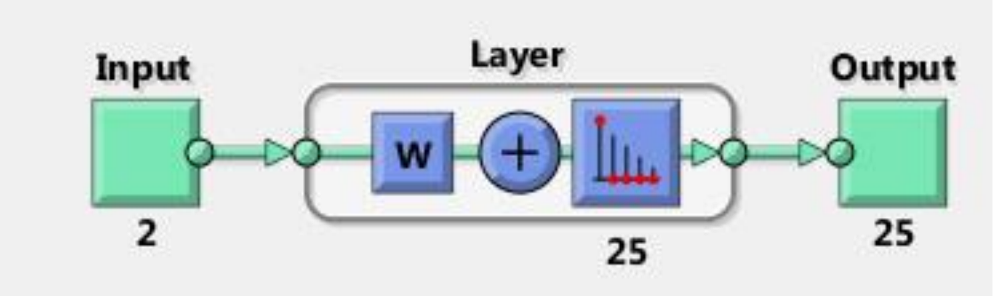


Figure-3.4 Plot of the Self-Organizing Map used to generate the topology.

Sequentially following through Figures 3.1 to 3.3 we can understand clearly the plotting of the SOFM points to the input space. However, Figure-3.4 can be attributed as the Input variables namely as the X, Y coordinate (the data points in Input_data1.txt) and the corresponding weight matrix, which is followed by a single layer of 25 neurons with an objective of producing the output of 25 such points in input space (shown in Figure-3.2).

Furthermore, the usage of MATLAB built-in functions helps to explain the above diagram explicitly.

net = newsom(PR,[D1,D2,...],TFCN,DFCN,OLR,OSTEPS,TLR,TND)

where PR - R x 2 matrix of min and max values for R input elements., Di - Size of ith layer dimension, defaults = [5 8].,TFCN - Topology function, default ='hextop'.,DFCN - Distance function, default ='linkdist'.,OLR - Ordering phase learning rate, default = 0.9.,OSTEPS - Ordering phase steps, default = 1000.,TLR - Tuning phase learning rate, default = 0.02;,TND - Tuning phase neighbourhood distance, default = 1.,and returns a new self-organizing map.

Here we have used the Di as [1, 25] which represents the single layer with 25 neurons, [D1,D2] are used as [-10,10] as this is our range of the numbers in our data set and all remaining are used with defaults.

After this network is built we set the number of epoch needs to calculate the convergence, which is 100. Now we train the network with the input vectors from the Input_data1.txt and finally we obtain the output in terms of layer weights. When these layer weights are plotted in SOM space the out topology can be visualized as depicted in Figure 3.5

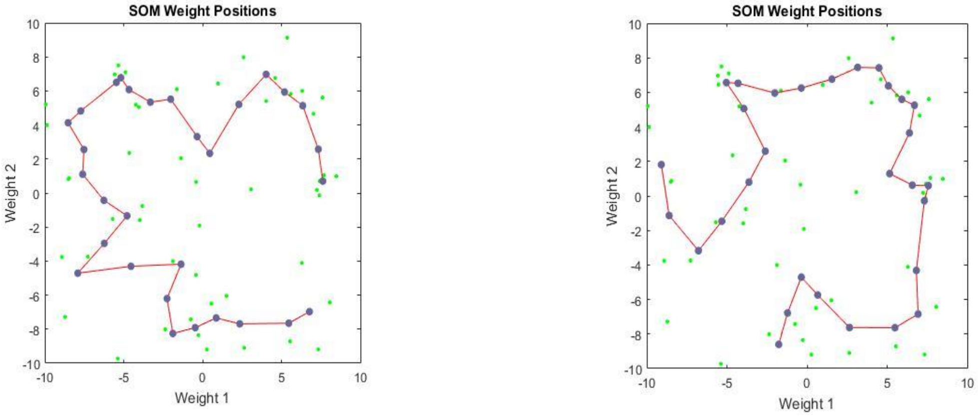


Figure 3.5 Plots the input vectors as green dots and shows how the SOM classifies the input space by showing blue-grey dots for each neuron's weight vector and connecting neighbouring neurons with red lines (right hand shows SOFM (KNN) and left hand shows SOFM (ANN))

In Figure-3.5 we can see the connecting blue dots with red lines passing through those parts which have a high concentration of points, so in case of failing nodes the nearest nodes can be considered without much searching the input space. Thus, we can see the coverage of the area is ensured for considering the range of nodes such as 20 nodes, 25 nodes, 30 nodes, etc.

3.6.2 The Simulation:

Due to the stochastic nature of the environment, a useful performance metric is typically not the result of a single experiment, but rather an average value, a minimum or maximum. Thus, a single function call of the optimizer algorithm can be very expensive. Other problems include no prior knowledge of error surface so that efficient error surface calculation cannot be determined. As a result, the number of experiments to be done is not known.

In order to overcome such problems and keeping the considerations of the target system we use PROWLER—PROBABILISTIC WIRELESS NETWORK SIMULATOR V1.25 with a test bed of 50 sensors placed in a matrix of 10 × 10 sq units, the positions of various sensors were recorded previously and supplied to the simulator as input.

The input spatial distribution of sensors under test consideration as viewed in MATLAB 7 and shown in Figure 1 is taken as the input topology file.

The topology file is modified according to the algorithm in Section 3.1. Step 2. Figure 3.2 describes the outcome.

Now using Step 4 of Section 3.1 we remap the circle coordinates to the cross-coordinates as shown in Figure 3.3.

From Figure 3.3 it is clear that due to some duplicate values the number of remapped values is less than what is expected. Now, by using Step 5 of Section 3.1, we can remove these duplicate values and finally obtain unique values from the input topology file. Now, as described in Step 7 of Section 3.1, we use PROWLER V-1.25 for simulating with Spanning Tree Protocol (STP), with the centre as root node.

The list of assumptions made while running the simulation on PROWLER-V1.25 [13] are as follows.:

(1) Each node has the following fields in the routing table.
xID: The identifier of the neighbour.

InLink: Quality of the directed link (*xID* → *ID*). *OutLink*: Quality of the directed link (*ID* → x*ID*).

Hop: the hop-number of mote *xID.*

Note: Each node is assigned a unique ID, hop number (initially NaN except where the root node is zero).

(2) Each node wakes up periodically and transmits its *ID*, hop number, and table data. Upon receipt of message from node *i*, node *j* updates its own table.
   (i) Updates the *InLink* property of *i*.
   (ii) Updates the *Hop* property of *i*.
   (iii) Updates the *OutLink* property of *i*, if the received table contains information about *j* (the *InLink* value is used).

(3) Each node transmits the table data with certain finite probability. The transmission probability is the function of the design parameter and the content of the table.
   (a) Initially $p = P/8$.
   (b) For all the nodes with a hop-number *NaN*, $p = P/8$.
   (c) If the hop-number of the node changes, p is set to P.

(d) If a mote *j* receives a message from node *i*, indicating that *i* has no information about *j*, but *j* has a good *InLink* property of *i*, then node *j* sets $p = P$.

(e) After each transmitted message $p = P/2$.

Using the above considerations, the Spanning Tree algorithm was run on the test bed; Figures 4-9 are the performance graphs obtained.

In order to interpret them we use a performance metric derived from the number of receiving motes. In the network, the more the motes receive the better and the consumed power which is proportional to the settling time, that is, in the time to build the Spanning Tree, the less power that is used the better.

Considering Figure 3.6, we have a very small region around $p = 0.3$ and $p = 0.6$, 1. The participating motes are around +90%, which shows maximum coverage over the test area and the settling time is around 0.25 seconds, which in terms of the energy metric is appreciable. But the steady state in the receiving nodes has not yet been achieved.

A similar explanation applies to Figure 3.7. But considering Figure 3.8, we find that there is a stable number of receiving nodes for $p = 0.1$ to $p = 0.8$ and the settling time is around 0.3 seconds, and considering Figure IX, we find that there is a stable number of receiving nodes for $p = 0$ to $p = 0.4$ and $p = 0.6$ to $p = 0.1$ and the settling time is around 0.35 seconds; both these cases are good examples of a simple trade-off between energy consumed and the mean receiving nodes. So, in regard to the input topology as presented, the saving in terms of nodes is 20% to 30%; these nodes can be used when the depletion of energy in other nodes occurs and hence the lifetime of the network can be increased. Though a saving of about 20%–30% is not enough for any sensing mission, this saved percentage could reduce the number of sensors to be deployed/used for the next time. The savings in terms of nodes is absent if we use all the 50 sensors at a time; the performance characteristics as depicted in Figure 3.10 show no steady region for mean receiving nodes versus p as in Figure-3.8 or Figure 3.9, but in terms of settling time Figure 3.9 shows the optimized settling time as compared to Figure 3.10.

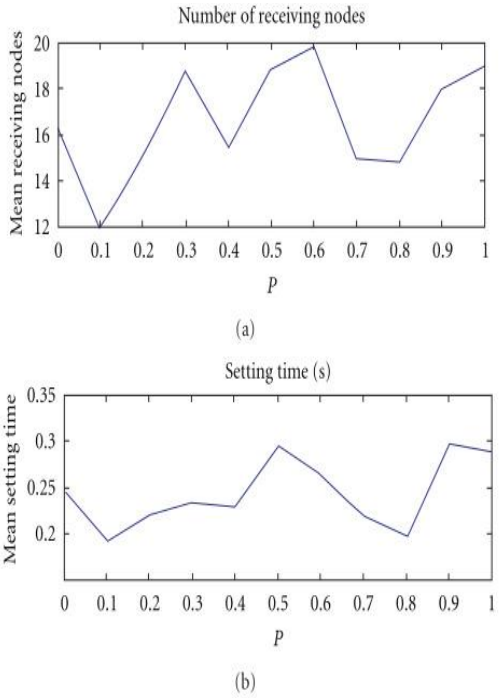


Figure 3.6 The performance graph with 20 neurons in the test condition.

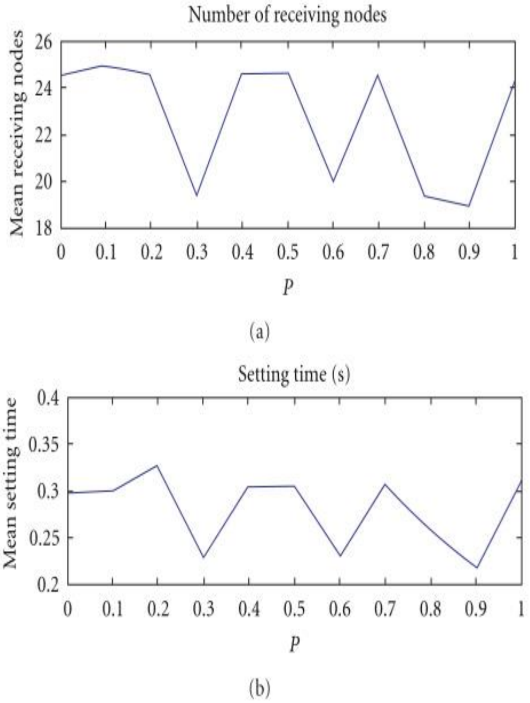


Figure 3.7 The performance graph with 25 neurons in the test condition.

Considering the case for Figure 3.7 we have the following node ID generated while running the simulation of PROWLERV-1.25 for transmission probability ranging from 0 to 1. Referring to the Figure 3.8, the node IDs from p = 0.1 to 0.8 can be used for information dissemination efficiently over the network covering the test area. The combination from p = 0.9 is omitted due to inadequate coverage because a lesser number of participating nodes are seen from Figure 3.8. Now, if we consider time

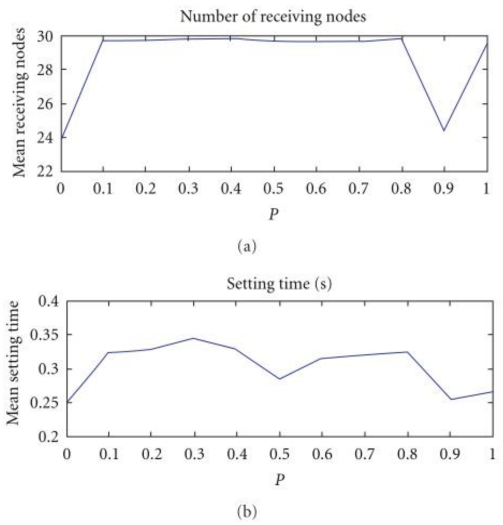


Figure 3.8 The performance graph with 30 neurons in the test condition.

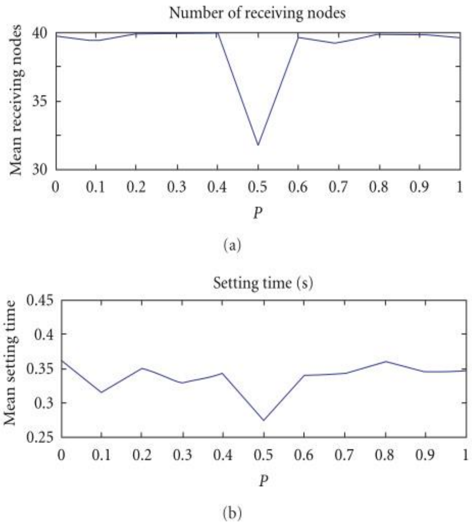


Figure 3.9 The performance graph with 40 neurons in the test condition

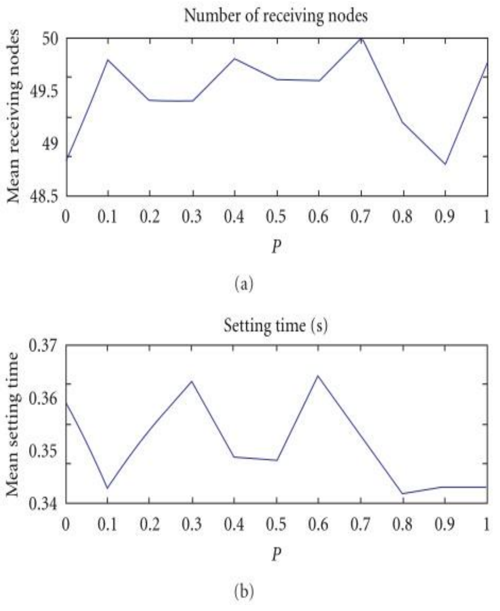


Figure 3.10 The performance graph without SOFM topology building algorithm in the test condition

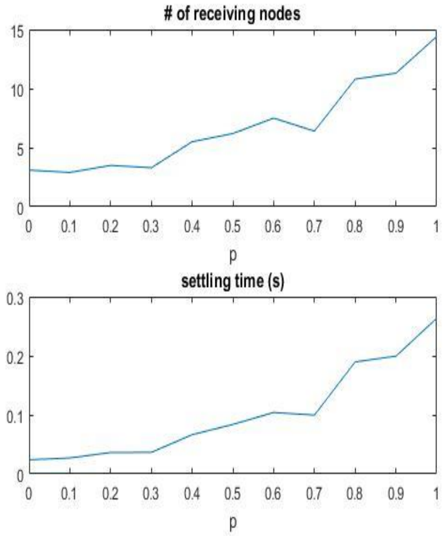


Figure A

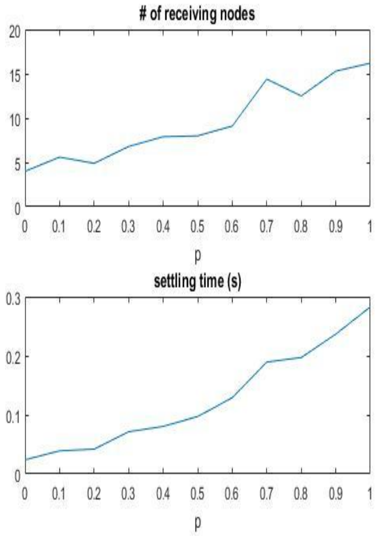


Figure B

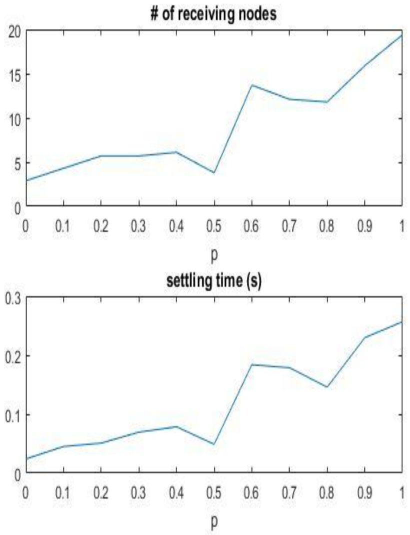


Figure C

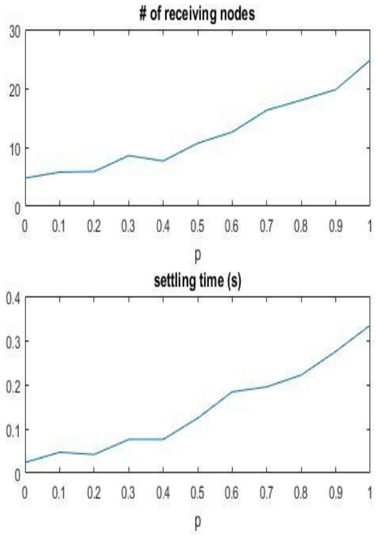


Figure D

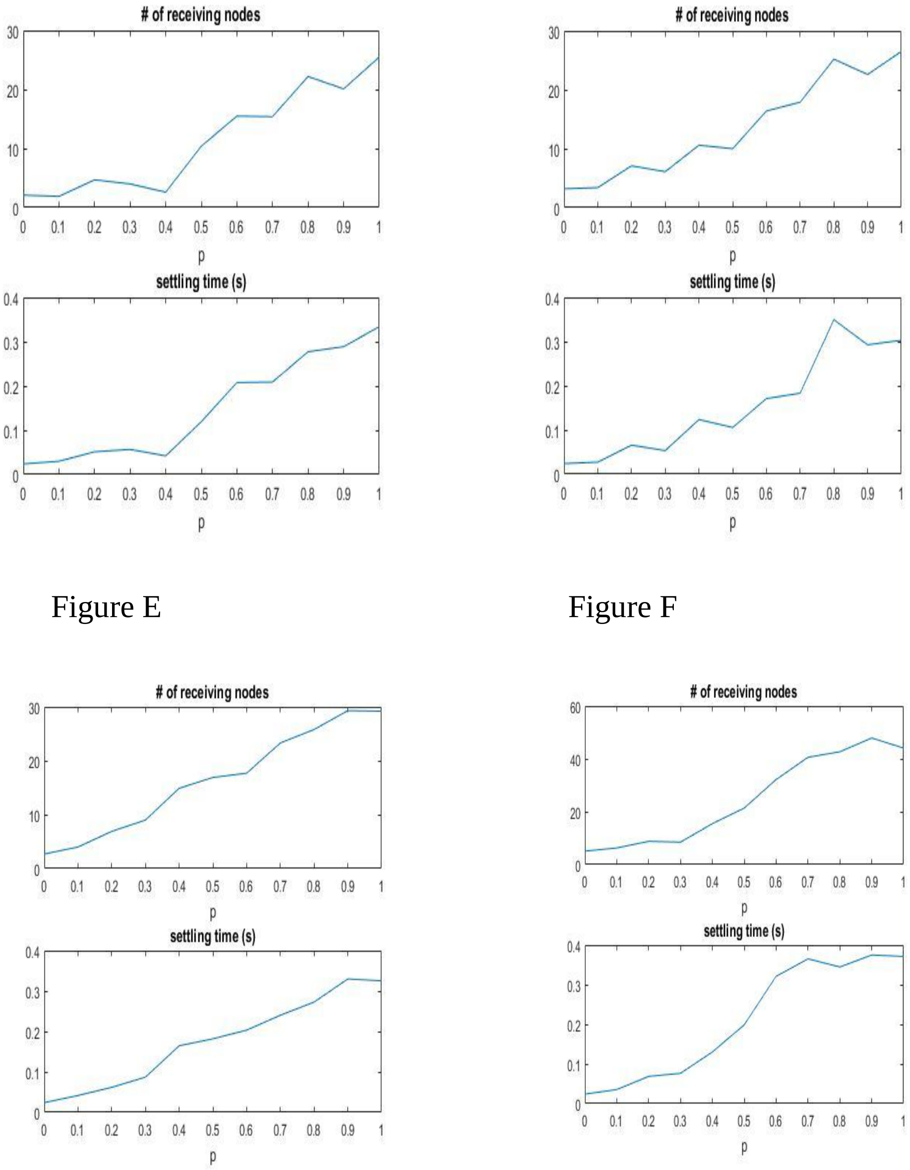


Figure E

Figure F

Figure G

Figure H

Figure 3.11 The Figures A through G show the performance of the SOFM (KNN) created topology for 20, 25, 30, 35, 40, 45 and 50 nodes. Figure H is performance without using SOFM (ANN). All the outputs refer to the usage of Flooding Protocol.

varying usage of the paths as shown in Table 1, as in each p being selected for 5 runs as done in our simulator, we would obtain the node usage (i.e., number of times a …….It can be easily seen that all the SOFM generated topologies do have the same nature of graph and each has the best coverage for the range of probability around 0.6 to 1. However, it can also be implied that the choice of considering a particular set depends upon the user and how long does he/she want the network to perform. It should be considered that the lower number of nodes at the start will prolong the network functioning for a long time. Another important observation is that by considering SOFM algorithm for 50 nodes it's seen that the maximum of 30 nodes are sufficient for covering the area and a saving of 20 nodes is observed in the flat sense. Rest 20 nodes will be used to back up the network in case of node failure. Figure H of Figure 3.11 shows the performance of the network without using SOFM in which the maximum number of nodes used to cover the topology is around 42, but by using SOFM this usage of the nodes drops down to 30. Similarly, Figure H of Figure 3.12 shows almost all the nodes are used to cover the area but by using SOFM of 50 nodes only 30 nodes are used to cover the network.

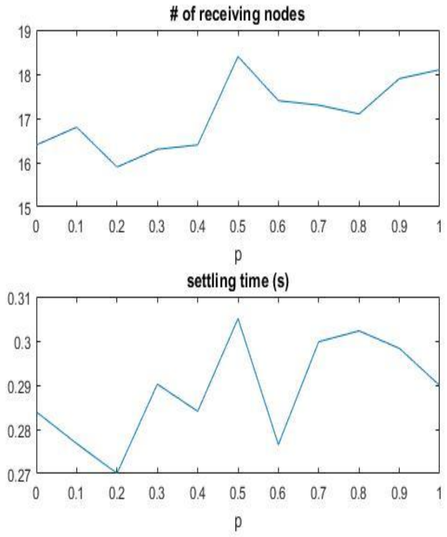


Figure A

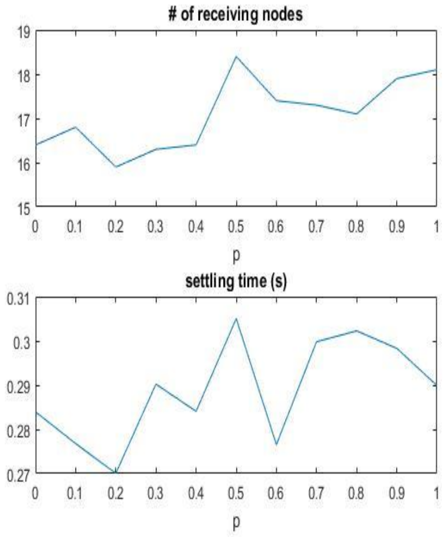


Figure B

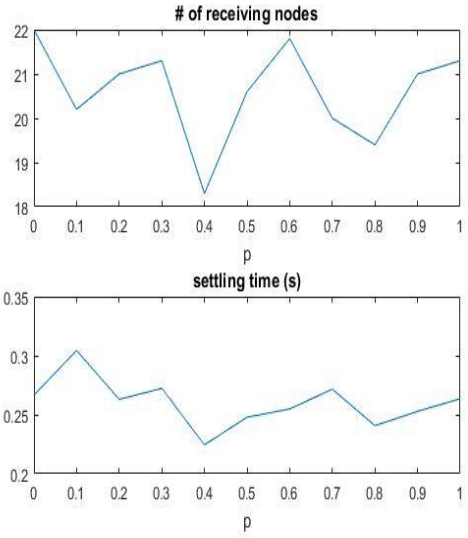


Figure C

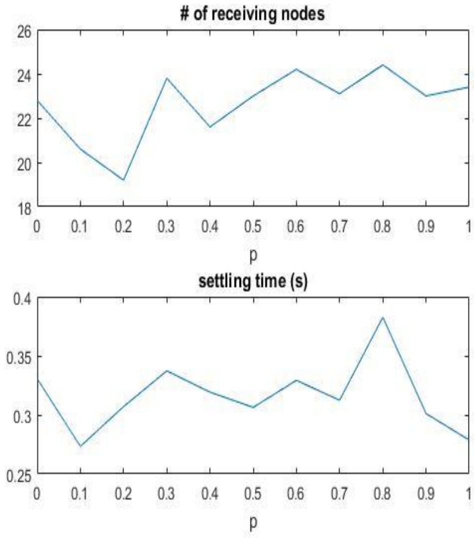


Figure D

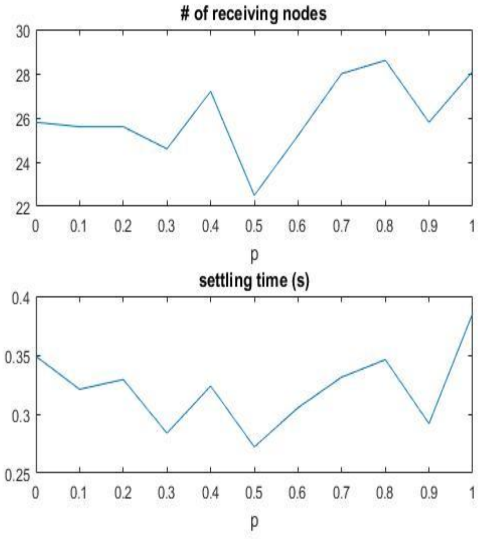


Figure E

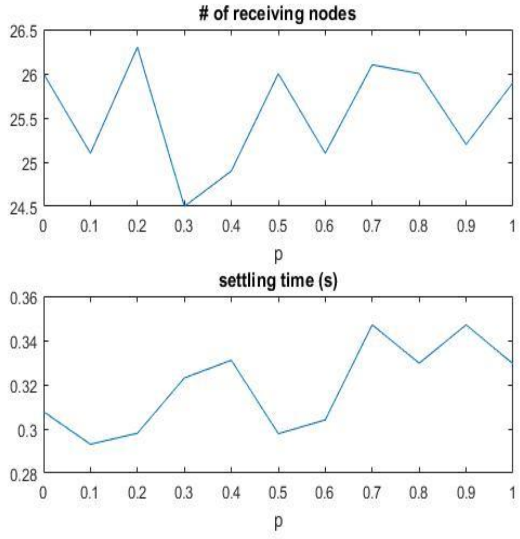
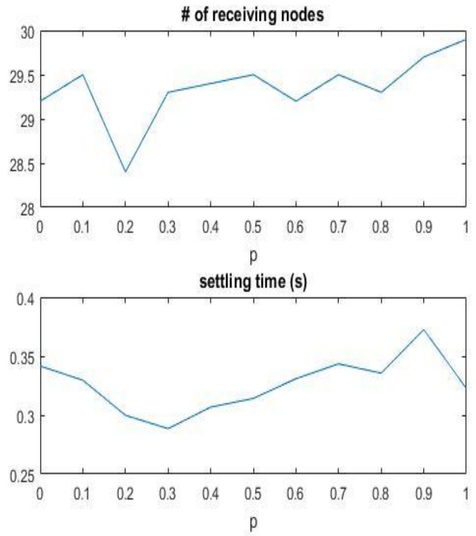


Figure F



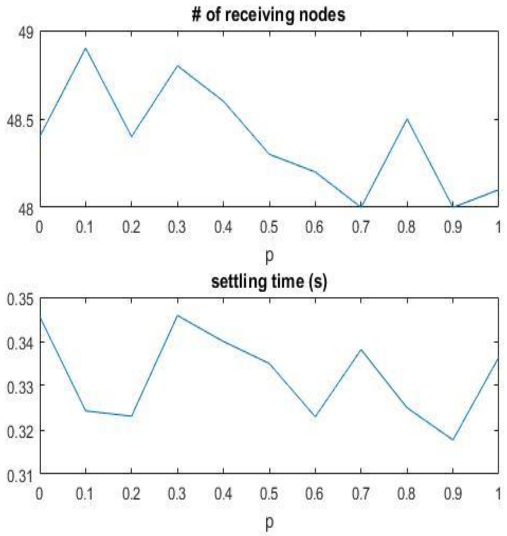


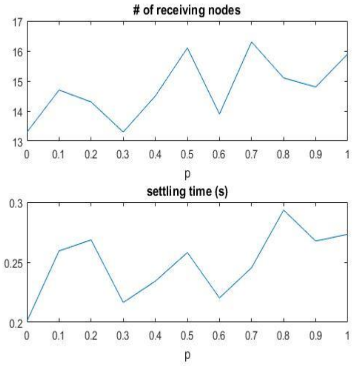
Figure G

Figure H

Figure 3.12 The Figures A through G show the performance of the SOFM (KNN) created topology for 20 ,25, 30, 35, 40, 45 and 50 nodes. Figure H is performance without using SOFM(ANN). All the outputs refer to the usage of Spanning Tree Protocol.

It can also be seen for Figure A through G of Figure 3.12 that there is a jittery state of the number of nodes covered in range probabilities, which is one of the difficulties of considering the above model as it will not be able to cover the network every time.

A similar caveat follows for SOFM (ANN), which is as depicted in the figure below:



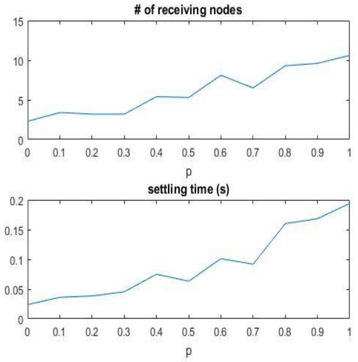


Figure A

Figure H

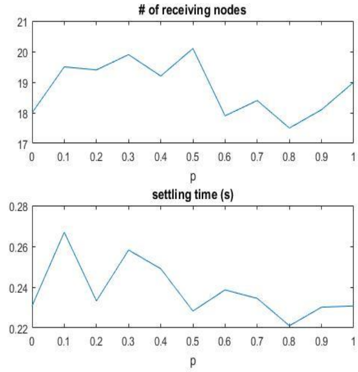


Figure B

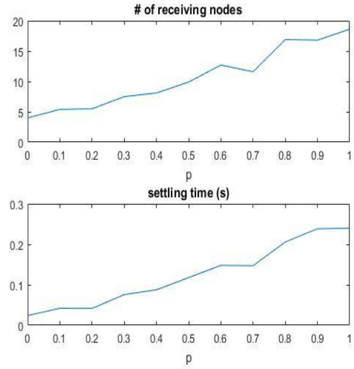


Figure I

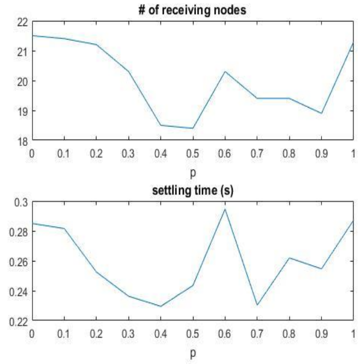


Figure C

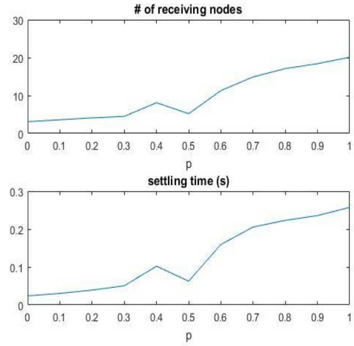


Figure J

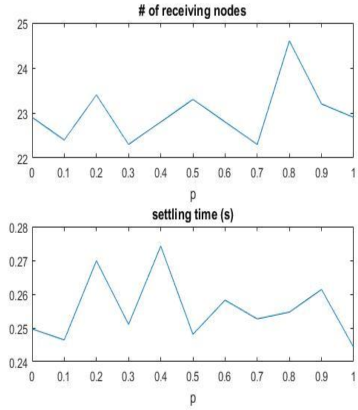


Figure D

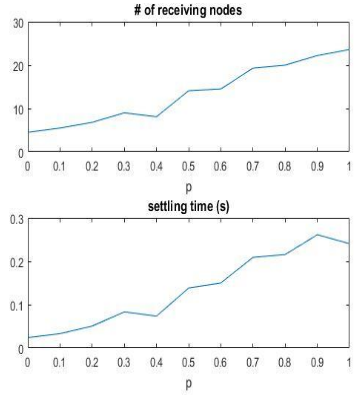


Figure K

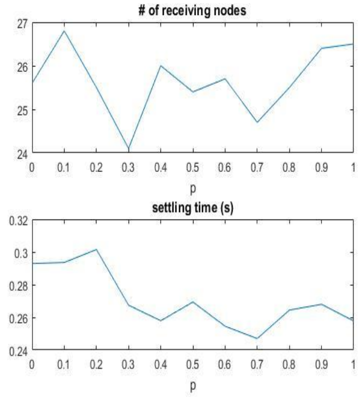


Figure E

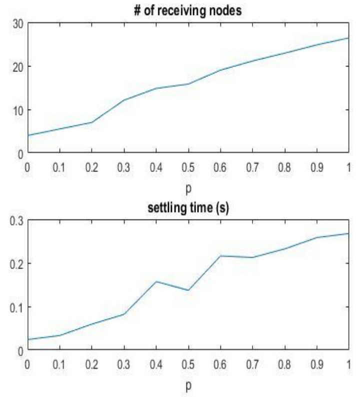


Figure L

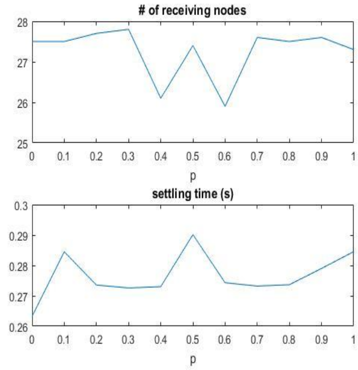


Figure F

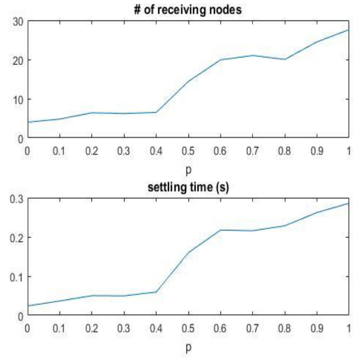


Figure M

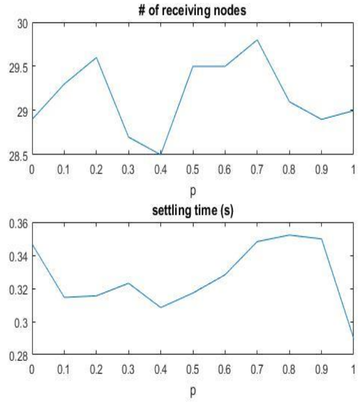


Figure G

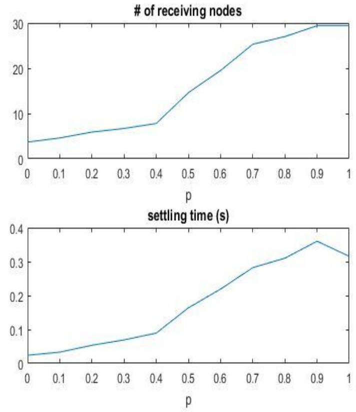


Figure N

Figure 3.13 This plot represents the performance of SOFM (ANN) for Spanning Tree Protocol (Figure A to G) and that of SOFM (ANN) for Flooding Protocol (Figure H to N)

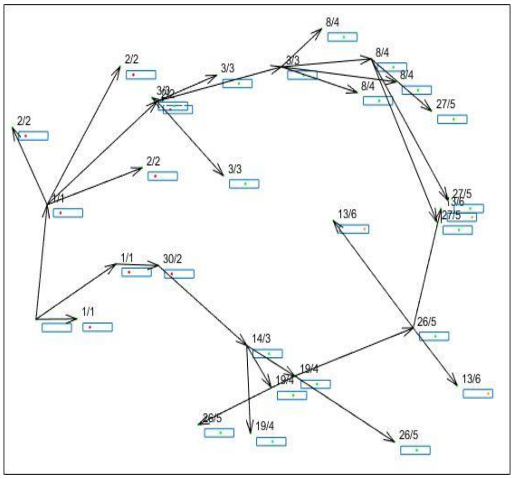


Figure A

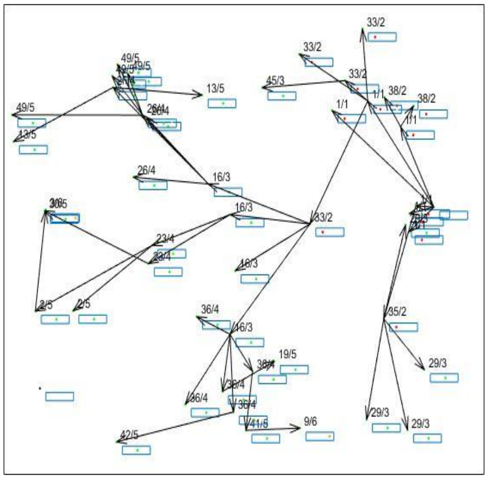


Figure B

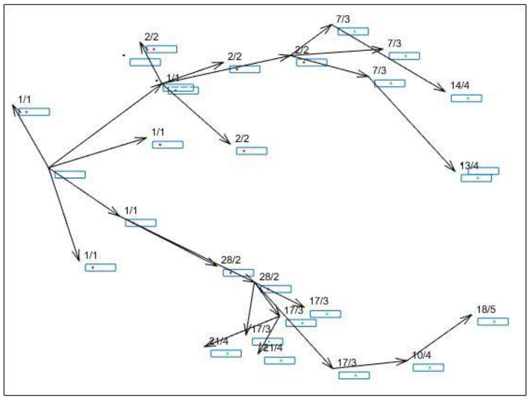


Figure C

Figure-3.14 This plot shows the topology generated using SOFM (KNN) (Figure A,) without using SOFM (Figure B) and SOFM (ANN) (Figure C). This plot refers to the execution of the Spanning Tree Protocol.

This Figure 3.14 (Figure A and C) clearly shows the pendent-like topology as predicted by Figure 3.5, whereas the one without the use of SOFM builds a haphazard structure and hence more nodes are required to cover the designated area. However, as a result these structures are not stable. This might not be the case when there may be another execution of the same algorithm with the same settings. In contrast, the SOFM generated topology is stable and will be the same with the same settings at any instant of time.

3.7. Results and Discussions

The random distribution of nodes is finally given a definite energy-efficient topology building algorithm; that is, a SOFM topology building algorithm. Though it may seem that the training and retraining required by SOMTB algorithm would require energy, this can be utilized to generate alternate routing paths very easily with low energy expense, which is not suitable for small pilot areas but effective for larger geographical areas.

During the simulation process the SOMTB algorithm which forms the backbone of the Spanning Tree Protocol here will have the sender node forwarding the routing table to its immediate neighbour. This immediate neighbour will update its own table, discarding the redundant information. Thus, each table will contain only the information about the immediate neighbours. This partial information in a larger sense will be accumulated and delineate the complete network picture. So, the routing table will remain manageable in terms of size and computation time.

Important advantages, in addition to low power consumption, include simplicity, inherent robustness to node or link failure, changing network geometry (in case of battery depletion), reduced redundant packet transmissions and implicit network reconfiguration. The only disadvantage is the need for sufficient sensor density to maintain network operation. Simulations conform well and illustrate the promise of SOFMTB algorithms for applications such as event detection and monitoring over a large distribution area. This algorithm can be extended for use with other protocols used for WSN.

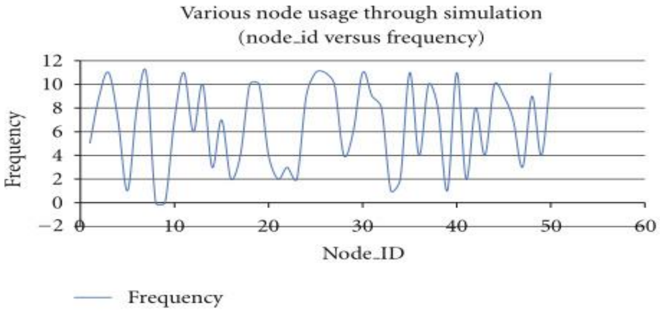


Figure-3.15: Various Node Usages through simulation (node ID versus frequency).

As it shows all the promise of efficiency and low power consumption, the implementation of synchronization should be done as future work.

Chapter 4

Energy Efficient Hierarchical Topology Building Schemes

In spite of the various constraints present in wireless sensor networks (WSN), their functionality has become popular across multiple application domains that require the transmission of information. Of these limitations, energy conservation is the most critical aspect, and a known strategies to save energy and to prolong the lifetime of WSNs is topology control. This chapter presents two schemes, a reliable energy-efficient topology control algorithm and a minimum transmission level algorithm, for use in wireless sensor networks. The reliable energy-efficient topology control algorithm considers the residual energy and number of neighbours of each node in terms of cluster formation, which is critical for well-balanced energy dissipation within the network. Within this, a knowledge-based inference approach is employed to select cluster heads, as reliable cluster heads maintain connectivity and coverage with respect to time. The algorithm thus not only balances the energy load of each node but also provides global reliability for the whole network. The simulation results demonstrate that the proposed algorithm efficiently prolongs network lifetime and reduces energy consumption. In the other scheme, a minimum transmission level algorithm for wireless sensor networks, the farthest node is determined based on local positional information, thereby eliminating the process of searching for the most distant node, which in turn saves energy.

4.1 Introduction

A wireless sensor network consists of a large number of tiny, low-powered, energy-constrained sensor nodes that nevertheless each have a full complement of sensing, data processing, and wireless communication components. The sensor nodes in WSNs are small battery powered devices with limited energy resources, and their batteries cannot be recharged once the sensor nodes are deployed. Thus, minimisation of energy consumption is one of the primary goals in the design of network topology. Clustering is one of the practical solutions that helps to prolong the lifetime of a network and hence provides network scalability [58]. In the process of clustering, the sensor nodes may form groups with respect to physical distances or energy or sensor types; these groupings can also be based on a linear combination of any of these properties. Each cluster has a cluster head, which represents the set of nodes in a cluster. A cluster head can collect data from nodes within the cluster before aggregating the data and forwarding it to the base station. If the data sent to the cluster head is reduced, then the overhead can also be significantly reduced in terms of energy, and bandwidth can thus be saved [41].

Extensive research has been devoted to ad hoc sensor networks in terms of using power control to reduce interference and improve throughput, as addressed in [188, 179]; topology control by tuning transmission powers as discussed in [82, 128, 130]; and the ways in which both IEEE 802.11 and Bluetooth support low-power modes [41, 65]. Ways to design low-power motes on IEEE 802.11-based multi-hop networks are also addressed in [172]. Among the many challenging issues these pieces of research raise, in this context, the fundamental and practical problems to be addressed are how to build an effective topology for the sensor nodes, and how to identify a node-deployment function that is optimal for sensors. In most current designs, random and uniform distributions are popular due to their simplicity. However, as will be shown later in this chapter, node deployment-related issues have a serious impact on system functionality and thus on the lifetime of sensor networks. Traditional random and uniform distributions are thus not suitable because of the sink routing-hole phenomenon. Instead, a non-uniform, power-aware node deployment scheme is proposed as the primary focus in this work.

Connecting features such as node energy constraints, node distribution, and scalability means that WSNs are inherently different from existing systems and networks. Unlike the patterns seen in traditional mobile ad-hoc networks (MANETs), in WSNs, one or several nodes (called a sink) are designated as a gateway between the networked system and users. Such sinks play a critical role in the network; their loss directly leads to failure of the entire WSN. Furthermore, not only the sink but also the direct neighbours of the sink are essential for network functionality. These are required to establish and maintain connectivity between the sink and more distant data nodes. When these set of nodes fail, no data can be delivered to the sink due to the lack of forwarding nodes, and thus the whole system fails. Sinks may be designed to be more powerful and may have permanent power sources to minimise failure rates. However, their nearby neighbour nodes are generally common sensor nodes with limited resources, making them vulnerable to failure by running out of energy. For most data gathering applications, the forwarding workload of sensors increases inversely with their distance to the sink. A node closer to the sink usually has a higher relay workload than that of farther away nodes. Accordingly, the direct neighbours of the sink have the greatest transmitting workload, and are thus likely to deplete their energy rapidly. When they fail, no matter how many remaining nodes are still active, there can be no communication with the sink. From the sink and the user’s perspective, the whole system fails if the sink is isolated from the rest of the sensor nodes. Multiple sinks or mobile sinks cannot eliminate this problem, as multiple sink routing-holes will be generated accordingly. Data funnelling and aggregation [123] techniques may alleviate the problem to some extent, but these cannot completely eliminate the problem The main objective of this work is thus to provide a long-term continuous connectivity scheme, attempting to address the problem by designing a power-aware topology management scheme. The majority of attention is thus paid to the connectivity of the network, as this is a prerequisite for its other purposes such as sensor coverage. Without a valid data path, an active node has the same role as a dead one, and any sensing area covered by unconnected nodes is inaccessible [5, 90]. Traditional approaches as described above do not take the differential loss of energy in different sub-areas into consideration, and consequently, while they may work well for some snapshots of the network such as the initial stages, they are unlikely to guarantee quality of service throughout the system lifecycle. This Chapter's contributions are thus mainly in two areas.

Approach I:

1. Production rules are mapped to create an adequate representation of a network topology.
2. A cluster-head selection mechanism is developed to consider the residual energy, number of neighbours, and centrality of each node, using production rules to facilitate cluster-head selection.

3. A reaoning mechanism, used to create a reliable multi-hop routing algorithm that creates efficient routes among cluster heads, is developed.
4. The resulting reliable energy-efficient two-tier routing protocol is implemented and evaluated through simulation.

Approach II:

1. Local information, such as the positions of nodes, is used to calculate the distances between the nodes.
2. As a result, the communication range of each nodes to its farthest neighbour is reduced so that all neighbours remain covered, reducing the energy required for the construction of the topology.

4.2 Background Ideas

4.2.1 Production Rules

The main reason for using production rules is the simplicity of achieving the desired goal for the system. The production rules used here were based on the parameters of the WSN under consideration. These parameters included the number of neighbours and the residual energy of each node under consideration. The justification for these considerations is as follows:

1. The Residual Energy, Er, represents the remaining energy of each node. The higher the Er, the more data that can be processed and transmitted, and the longer the lifetime that can be expected for the node;
2. The number of neighbours, N, affects proper cluster head election. It is reasonable to select a cluster head in a region where the node has more neighbours to facilitate message transference.

Before firing, W1 and W2 are set as weight values; the associated truth degrees are a1 and a2 respectively for Er and N; after firing, physically and logically permissible combination rules can thus be formed.

Rule 1.

$$A = a_1 * W_1 + a_2 * W_2$$

Rule 2.

$$a_1 = A * W$$

$$a_2 = A * W$$

Rule 3.

$$A = \max(a_1 * W_1, a_2 * W_2)$$

In terms of a_1 and a_2, the truth values of Er and N may be defined as

$$a_i = \frac{Present\ Value}{Max\ Value}$$

Thus, the a_1 truth value may be defined as

$$a_1 = \frac{Present\ residual\ energy\ of\ the\ Sensor\ node}{Unused\ sensor\ node\ energy}$$

Similarly, the a_2 truth value can be obtained per Kleinrock et al:

$$a_2 = (Number\ of\ the\ neighbors\ of\ the\ Sensor\ node) = 6$$

A is defined as the Chance function which maps places to real values from zero to one. The higher the value of A, the higher the preference for the node becoming the cluster head.

4.2.2 Minimum Common Transmission Level

The assumptions for executing the minimum common transmission level are as follows:

1. Each node has exactly one parent.
2. The QoS features are affected by reducing the redundancy caused by clustering.
3. The energy wasted or precipitated by unnecessary transmissions is cut off due with avoidance of redundant clustering.

4. As the connectivity between the nodes is based on a limited value of communication radius, message transmission is error free, and the noise value from to signal strength drop is minimised.

4.3 Assumption for the Algorithms

4.3.1 Assumptions due to Production Rules

The considerations of the cluster head selection algorithm are as follows:

1. The nodes do not know their positions or orientation; hence, the prior idea of topology is unknown.
2. All nodes are located in a two-dimensional space and each has a perfect communication coverage disk.
3. Every node starts in an unvisited site
4. The sink is the initiator of the process, and thus has a significant amount of energy such as a base station.
5. The time t_i of the i^{th} node broadcasting the CH (Cluster Head) msg (message) is $t_i = A_i * T$ where T = predefined max time allowed for CH competition.
6. When the cluster heads change after time T +D, where D= the interval between two runs of CH selection algorithm, to connect the cluster heads a connected dominating set algorithm must be used to form a reliable multi-hop connection.
7. There is no packet loss at the Data Link Layer.

4.3.2 Assumptions due to Minimum Common Transmission Level

The proposed protocol is a hierarchical topology construction protocol based on the growing tree technique. The idea is to illustrate some common message exchange sequences, to explore the use of timeouts, and to note how modification of the status of the node modifies the execution of the protocol. The protocol is based on the following assumptions

1. The nodes know their positions but have no list of neighbours.
2. Every node starts in an unvisited state.

3. The sink is the initiator of the process.
4. The protocol ends when every node is in active mode or in sleeping mode.

4.4 Algorithm Descriptions

4.4.1 Algorithm from Production Rules

```
INPUT: the cluster head election chance Ai of each node Ni.
OUTPUT: the set of cluster heads.
Step 1: Set the timer $t_i$ of the node $N_i$
Step 2:
While (ti is not expired)
    if (Ni does not receive CH msg)
    Ni broadcast CH msg to neighbors
            else if (Ni receive 1 CH msg from Nj)
                    Ni selects Nj as CH
                    Ni transmits the Msg JOIN to Nj
            else if (Ni receives m CH msgs from other
                    m nodes)
            from m nodes Ni selects Nk with highest
            cluster head election chance as CH
            Ni transmits the msg JOIN to Nk
    End if
End While
```

Cluster head Selection Algorithm

After the clusters are formed, it is important to connect the cluster heads to aggregate the data and forward it to the base station through a multi-hop path. As the cluster heads change over time T, a connected dominating set algorithm must be used to form a reliable multi-hop connection among these cluster heads.

1. Calculate d, xi(a) (maximum degree of neighbours of distance 2, a value set by the primal-dual algorithm [11])

2. Become a dominator by going to the dominating set with probability pi = min(1;xi(a):ln(d+1))

3. Send status (dominator or not) to all neighbours

4. If no neighbour is a dominator, then declare as a self-dominator

Cluster head Connecting Algorithm

4.4.2 Algorithm from Minimum Common Transmission Level

Below is the list of the steps for the algorithm thus produced :

Step 1

1. The sink node initiates the protocol by sending a Hello message to all the nodes in its communication range. A Hello message includes the node's address and level (number of hops from the sink); in the case of the sink, this is equal to 0.
2. The sender node programs a timeout to stop listening for answers from its neighbours.

Step 2

1. All unvisited nodes within the transmission range of the sender node that received the Hello message answer back with a Reply message to set the sender as their default gateway, changing their status to In-Process mode; previously visited nodes will not respond.
2. The receiver nodes calculate their distances from the sender node and send these as the metric within the Reply messages.
3. The receiver nodes simultaneously set a timeout in case they do not receive an acknowledgment from their default gateway, which is now the sender node.

Step 3

1. Once the timeout of the sender expires, the node checks the list of neighbours that answered with a Reply message
2. If the sender node did not receive any answers, it turns itself off and changes its status to Sleeping mode.

3. If the sender node received at least one answer back, it goes into Active mode and

i. Adds the neighbours as its Children.

ii. Sorts these children based on their metrics.

iii. Reduces the communication range of the node to the distance between it and its farthest child.

iv. Declares itself as Parent and sends a unicast message to each child in the sorted order to let them know they were selected.

Step 4

1. Once a node receives a confirmation message from its default gateway, it waits for a random amount of time before sending its own Hello message to discover new unvisited nodes.

2. If the timeout of a node in In-Process mode expires, it means that the node was not selected; it turns itself off and goes into Sleeping mode

4.5 Simulation and Results

The above protocol was simulated using Atarraya [60], a generic, Java-based, event-driven simulator for topology control algorithms in wireless sensor networks. The advantage of using Atarraya is its GUI, which is based on the real-time display of the network during execution of the routing protocol. In figure 4.1, the testing and comparison of the new protocol is displayed with a standard Energy Efficient Connected Dominating Set (EECDS) [182] protocol used as a reference.

The Energy Efficient Connected Dominating Set (EECDS) algorithm creates a maximal independent set in the first phase and then selects gateway nodes to connect the independent sets during the second phase, allowing them to connect and form a connected dominating set. The main disadvantage of the EECDS algorithm is its message complexity. In both phases of the algorithm, competition is used to determine the best candidates to be included in both the independent sets and the final tree. This process is very costly in terms of message overhead, because each node must consult its neighbours for their status in order to calculate its own metric. The execution stages of the new protocol do not require this phase; thus, it inherits all the advantages of the simple, yet robust features of other simple routing protocols. Figure 4.2 depicts how the energy of the modified EECDS algorithm by common minimum transmission level acts to judiciously spend energy to build the topology with additional cluster heads. Similarly, the A3 [121] belongs to a family of protocols which use the “growing tree” technique to build a connected dominating set; here, connectivity is the main objective. The Reliable Two Tier Energy Efficient (RTTEE) algorithm uses production rules to select the cluster heads as the first tier and CDS based cluster head connections as the second tier. In Figure 4.2, a comparison between the RTTEE algorithm and A3 can be seen: this shows that RTTEE algorithm performs better in terms of energy saving, but that the number of dead nodes with respect to time is higher due to higher operational energy thresholds. These algorithms thus seem to compensate for one another in Figures 4.2 and 4.3 respectively.

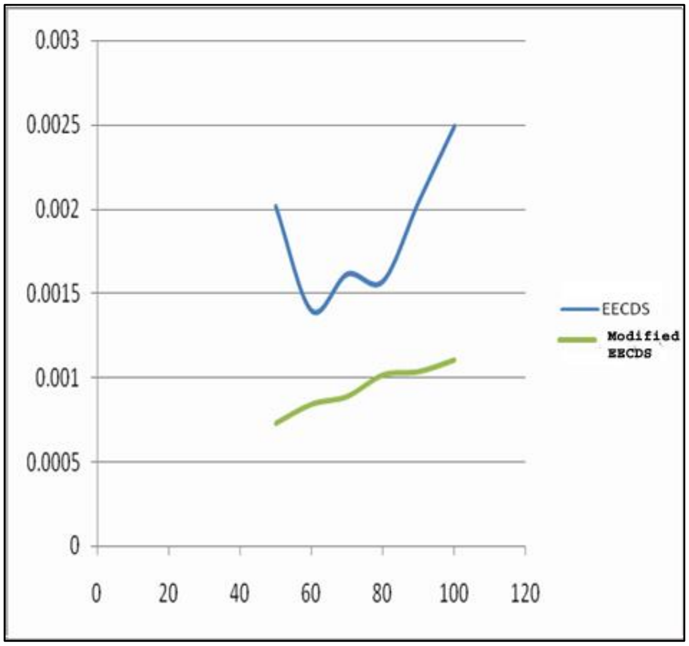


Figure 4.1 Number of Nodes versus Energy Spent ratio using Algorithm from Minimum Common Transmission Level

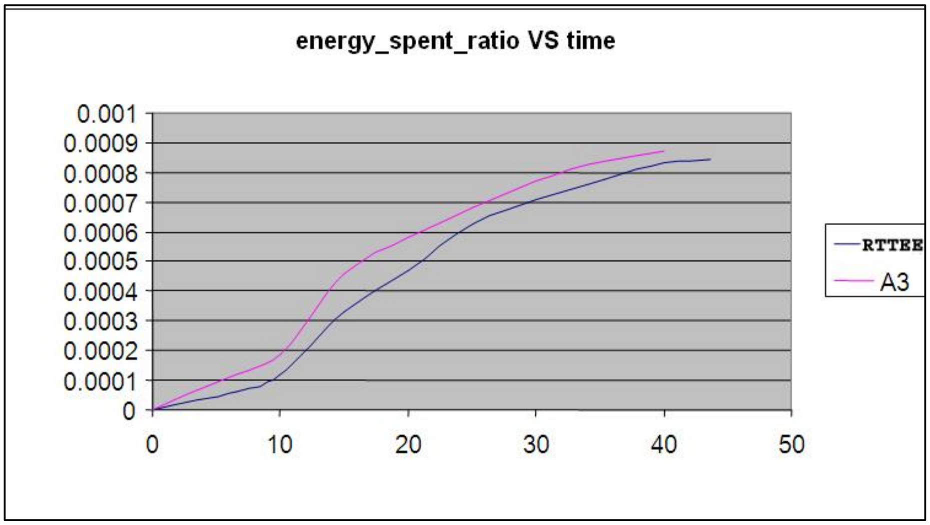


Figure. 4.2 Energy Spent Ratio versus time comparison between A3 and RTTEE algorithms

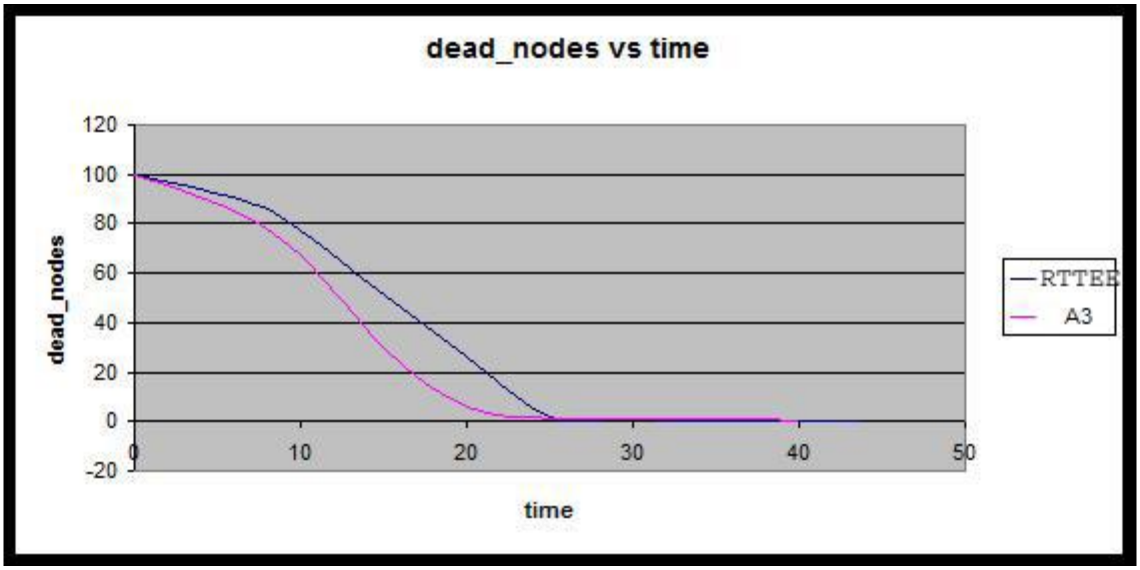


Figure 4.3 Number of dead Nodes versus time comparison between A3 and RTTEE algorithms

4.6 Conclusions and Future Scope

There remains huge scope for working towards the development of the relevant protocols using neural networks, fuzzy nets, genetic algorithms, and similar methodologies. However, the protocols as examined already throw some light onto the question of energy efficiency factor by using totally different methods. Future work must therefore be done on testing the above protocols in large scale sensor networks to mimic the current thrust of the development of advanced wireless sensor networks.

Chapter 5

Topology Management and analysis using Network Evolution Model

In the study of Wireless Ad hoc and Sensor Networks, clustering is a significant research problem as it aims at maximizing network lifetime and minimizing latency. A large number of algorithms have been devised to compute ‘good’ clusters in a WSN. In this chapter, by using the concept of local world model the natural functional mechanism of clustering in wireless sensor networks is deduced analytically which resulted in the relation between the numbers of connections of a node to the probability of clustering. This theoretical framework based on a complex network model is found to have reasonably matched with measures obtained by simulation studies conducted on tree-based clustering algorithms.

5.1 Introduction

Wireless sensor networks (WSNs) are made up of a large number of sensor nodes. These nodes are usually deployed in the environment to monitor several physical phenomena. However, sensor nodes heavily depend on batteries as they are the only source of energy in many WSN applications. As a result, one major problem in WSNs is known as topology management that leads to energy efficient transmission of data. In this regard, connections are set with nodes that are close enough for a radio signal to arrive with acceptable signal strength. However, to improve energy efficiency, topology control process helps in reducing the connections with other neighbors of the node in the network. Topology control is an insistent process in which there is an initialization phase ,which is common to all WSN deployments. In the initialization phase, nodes make use of the revelation process by using maximum transmission power to build the initial topology. The initial network topology includes of connections and nodes that allow direct communication and every node communicates with a subset of the nodes according to the distance between them.

Often, the topology of a large wireless network is structured regarding a hierarchy where the network is viewed as some clusters, and in each cluster there is a cluster head and other normal members. Normal members in a cluster communicate only with the cluster heads and the cluster heads communicate with the sink in one or multi-hop manner. There are many challenges in finding out the “best” set of cluster heads in a given network and in many formulations; these problems turn out to be intractable. Consequently, there are many algorithms to select the cluster heads and the clusters in a WSN that minimizes latency and maximizes network lifetime.

It has seen that the traditional network Topology Management in WSN is influenced by the concept of UDG (unit disk graph) [162, 74]. This idea of UDG actually reduces the given topology into Flat Networks and Hierarchical Networks with clustering [74].In Flat networks, all the nodes are considered to perform the same role in topology and functionality. Some of the well-cited examples are

Euclidean minimum spanning tree (MST)[4], cone-based topology control algorithm (CBTC),k-nearest neighbor (KNN)[18],directed relative neighborhood graph (DRNG)[93],TopDisc[37] and local Euclidean minimum spanning tree (MST)[94].

TopDisc discovers topology by sending query messages and describing the node states using three or four color system. It is a greedy approximation method based on minimum dominating set. In EMST or LEMST, each node builds its overall or local minimum spanning tree based on Euclidean distance and only keeps nodes on a tree that is one hop away from its neighbors. In DTG, a triangle formed by three nodes u,v, and w belongs to topology if there are no other nodes in the scope of the triangle. In KNN, a node sorts all other nodes in its transmission range in Euclidean distance and then links the k-nearest nodes as neighbors in the final topology. It is a scalable, parameter-free in WSNs, and effortless to implement. In DRNG, a link connects nodes u and v if and only if there does not exist a third node w that is closer to both u and v in the distance. CBTC uses an angle α as a key parameter. In every cone of angle α around node u, there is some node that u can reach.

. In many cluster head selection algorithms, every node is selected as the cluster head in different rounds and the probability of selecting a node as a cluster-head is the same for all nodes. In this method, the chances of energy dissipation in cluster heads reduces if we consider large homogeneous WSNs. There are other approaches where the idea of dominating set of graphs is used to devise an algorithm. Some of these methods are tree based. Most of these algorithms are distributed and they work with local (at the most 2-hop) information available from any given node. In case of hierarchical networks with clustering are homogeneous in functionality as cluster heads or cluster members. LEACH [105] is an excellent example of this kind in which the clusters are constantly updated . On the similar lines like the TEEN [3], APTEEN [97] , PEGASIS [58] are evolved mainly by the LEACH. In these protocols, the topology is updated through rounds as the energy of the sensor nodes gets depleted. However, LEACH uses a probabilistic model to select the cluster head for each round. In this chapter, we have considered three such topology construction protocols, namely simple tree, CDS-Rule K, and A3 protocols for further exploration

All the models described above shows a high degree of distribution for the nodes, which implies to present homogeneity in the graph properties [34].

Mathematically one can consider the unit disk graph case, in which we can assume that all nodes are randomly distributed in region S. Each node is positioned in a particular subarea with independent probability $\boldsymbol{\varphi = \frac{\pi r^2}{S}}$ where r is the transmission range. The probability that a subarea has k nodes is given by the binomial $\boldsymbol{p(k) = C_k^n \varphi^k (1-\varphi)^{n-k}}$ where n is the total number of nodes in the network. With the increase of n, this probability becomes the Poisson distribution $\boldsymbol{p(k) = \frac{(n\varphi)ke^{-n\varphi}}{k!}}$ then the average number of neighbor nodes is close to nφ−1.This shows that the UDG model has high concentration of connections when in large network that might promote excess energy consumption for the periodic topology maintenance and route selection process. Therefore, this is an inefficient way of topology construction. Hence one can consider other efficient option like scale free concept of analyzing the network when large number of nodes are considered.

Based on the above mathematical logic the application of Scale-Free Network concepts are used in the development of understanding the topological structure, functions and dynamical properties of wireless ad hoc networks. One of the most important models that can be used to characterize clustering algorithms formally is known as B-A model [2]. This model is based on two foundational mechanisms: growth and preferential attachment. A new node is added to the network at each step and connects with an existing node with a specific probability, which is related to the degree of the current node. The B-A network has the scale-free property and follows the power-law distribution.

The B-A network model is capable of capturing some underlying mechanism that is responsible for the power-law degree distribution. Still, it had many limitations. Li-Chen model [168] has improved upon B-A model. This model has been able to capture better the dynamics of networks constructed with a local preferential attachment mechanism.

The local preferential attachment model [159] is based on common sense that people can collect information quickly from their local community than from far away environment. Using preferential connection as the fundamental basis, many variations of the scale-free network model has been proposed during recent years such as comprehensive multi-local-world model [47]. Similar to preferential attachment model, the physical position neighborhoods model [55] mimics the actual communication network. The Poisson growth model [147] uses the number of edges added at each step as a random variable that corresponds to Poisson distribution. This model can generate many types of networks by controlling the random number.

Chen et al. [86] have studied an evolving mechanism for formalizing fault-tolerant communication topology among cluster heads with complex network theory. Based on the B-A model's growth and preferential attachment mechanisms, they not only used a local-world strategy for the network when a new node was added to its local-world but also selected a fixed number of cluster heads in the local world, to obtain a good performance regarding random error tolerance.

Luo et al [171] performed theoretical analysis and conducted a numerical simulation to explore topology characteristics and network performances with different energy distributions among nodes. Their results have shown that the network is better-clustered and average path length for transmitting data is reduced when energy distribution among nodes is more heterogeneous.

In [171], a new dimension is added as the nodes are not only allowed to join the network through preferential attachment but they are also allowed to leave the network or not join the network through non-preferential attachment. Further, the nodes distinguish themselves as cluster head nodes and normal nodes which is consistent with the function of many clustering algorithms in WSNs.

In this chapter, a formalism for some algorithms which compute clusters in a WSN using a modified local network model based on similar models proposed in [168], [136] and [171]. In particular, we have used three tree-based clustering algorithms, namely, simple tree, CDS Rule K and A3. Using our theoretical framework, we have also tried to quantify some of the observed features of these algorithms such as a number of cluster heads, the average degree of the resultant graph. The theoretically obtained measures have reasonably matched with measures obtained by simulation studies.

The chapter is organized as follows. Section 5.2 describes a very brief review of the local world model. In Section 5.3, we have defined the application of local world model to provide a framework to explain the functioning of three clustering algorithms. In Section 5.4, results of theoretical results and simulation studies are jointly presented. Section 5.5 concludes the Chapter.

5.2. A Brief Review of Topology Control Protocols, Scale-Free Networks and Local World Network model

In this section, we shall discuss very briefly the Li-Chen model [168] and two topology control protocols which are used in this chapter.

5.2.1 A3 Protocol [163]

The A3 protocol uses four types of messages: Hello message, Children recognition message, Parent recognition message and sleeping message. The sink node starts the protocol by transmitting an initial hello message to its neighboring nodes. Nodes accept the message if another node has not covered them; they set their states as covered, select the transmitter as its parent node and answers back with a Parent Recognition message. If a parent node does not get any Parent recognition messages from its neighbors, it turns off. The parent node sets a timeout period to accept answers from its neighboring nodes. Once the timeout occurs, the parent node sorts the list of neighbor receiving its message in decreasing order of some selection metric. Then, parent node broadcasts a children recognition message that includes the complete sorted list to all its candidates. Once the children accept the list, they set a timeout period proportional to their position on the candidate list. During that timeout nodes, they wait for a sleeping message from their brothers. If a node accepts a sleeping message during the timeout period, it turns itself off.

5.2.2 CDS Rule k Protocol [72]

The CDS Rule k algorithm utilizes Connected Dominating Set algorithm and pruning rules. The idea is to start from a big set of Dominating nodes that produces a minimum criterion and prune it according to a particular rule. In the first stage, the nodes will interchange their neighbor databases. A node will remain progressive if there is at least one pair of separated neighbors. In the second stage, a node chooses to unmark itself if it determines that marked nodes cover all its neighbors with higher precedence, which is given by the degree of the node in the tree. Lower level implies higher precedence. The ultimate tree is a more compact version of initial one with all redundant nodes with higher or equal priority removed.

5.2.3 Scale-Free Networks

Graphically [2, 83] the main difference between a random and a scale-free network comes in the tail of the degree distribution, representing the high-k region of p(k) . To illustrate this we have a plot Figure-5.1 between k and p(k) in log-log form which shows the following properties

1. For small k the power law is above the Poisson function, indicating that a scale-free network has a large number of small degree nodes, most of which are absent in a random network.
2. For k in the vicinity of the Poisson distribution is above the power law, indicating that in a random network there is an excess of nodes with degree $k \approx \langle k \rangle$.

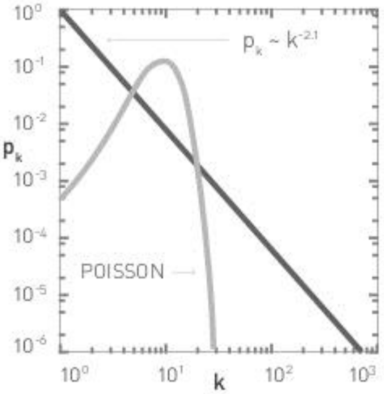


Figure 5.1 The log – log plot of p(k) vs k (grey stands for possion distribution ;black stands for power law)

3. For large k the power law is again above the Poisson curve. This is particularly due to the higher probability of finding a higher degree of node. This node may be some times called as HUB. This hub is several orders of magnitude higher in a scale-free than in a random network.

Mathematically [2] to understand the meaning of the scale-free terminology we need to find the moments of the degree distribution.

The expression for the nth moment degree of distribution is given by

$$\langle k^n \rangle = \int_{k_{\min}}^{k_{\max}} k^n\, p(k) \mathrm{dk} = \mathrm{C}\frac{k_{\max}^{n-y+1} - k_{\min}^{n-y+1}}{n-y+1} \text{--------(1)}$$

the lower moments have their own values as

1. When n=1: The first moment is the average degree, $\langle k \rangle$.

2. When n=2: The second moment, $\langle k^2 \rangle$, helps us calculate the variance $\sigma^2 = \langle k^2 \rangle - \langle k \rangle^2$, measuring the spread in the degrees. Its square root is standard deviation.

3. When n=3: The third moment, $\langle k^3 \rangle$, determines the skewness of a distribution, telling us how symmetric is p(k) around the average $\langle k \rangle$.

Generalization of the equation1 would lead to the following consequences [51]

While typically $k_{\min}$ is fixed, the degree of the largest hub, $k_{\max}$, increases with the system size Hence to understand the behavior of $\langle k^n \rangle$ we need to take the asymptotic limit $k_{\max} \rightarrow$

∞ in probing the properties of very large networks. The above equation predicts that the value of $\langle k^n \rangle$ depends on the interplay between n and γ

1.If $n - \gamma + 1 \leq 0$ then the first term on the r.h.s. of the above equation, k_{max}, goes to zero as k max increases. Therefore all moments that satisfy $n \leq \gamma - 1$ are finite.

2.If $n - \gamma + 1 > 0$ then $\langle k^n \rangle$ goes to infinity as k max $\rightarrow \infty$. Therefore all moments larger than $\gamma - 1$diverge.

For many scale-free networks the degree exponent γ is between 2 and 3 . Hence for these in the $N \rightarrow \infty$ limit the first moment $\langle k \rangle$ is finite,but the second and higher moments, $\langle k^2 \rangle$, $\langle k^3 \rangle$, go to infinity. This divergence helps us understand the origin of the "scale-free" term.

Figure-5.2 depicts graph between the the average number of neighbors versus standard deviation of the numbers of neighbors is plotted to give the visual of the different real world networks .This evidence comes not only from better maps and data sets but also from the agreement between empirical data and analytical models that predict the network structure [21, 26].

5.2.4 Local Network Model

We have used Li-Chen Model [168]. This model is used to form a generalized local world model. Using the generalized model, we have analyzed clustering algorithms of wireless sensor networks. In this model, each node has only local connection information. Nodes connect only in their local world based on their local connectivity. The following parameters are required to explain the dynamics with reference from Figure 5.1.

1. We start from a small number of nodes $\mathbf{m_0}$ and grow at each time step **t**.

2. When a new node chooses a connection to other nodes, the probability, $\prod \boldsymbol{k_i}$, that a new node is connected to a node ***i*****,** depends on the degree $\boldsymbol{k_i}$ of node ***i***. This probability is defined as follows.

$$\prod(\boldsymbol{k_i}) = \frac{\boldsymbol{k_i}}{\sum_j \boldsymbol{k_j}}$$

3. We select *M* nodes randomly from the existing network which is referred to as the local world of the new node.

4. When a new node arrives, we add that node with *m* edges, linking the new node to *m* nodes in the local world determined in (3) using the preferential attachment with a probability $\prod_{\mathbf{local}}(\mathbf{k_i})$**.** This probability is defined as follows at every time step **t**.

$$\prod_{\boldsymbol{local}} \boldsymbol{k_i} = \prod{}' \frac{\boldsymbol{k_i}}{\sum_j \boldsymbol{k_j}}$$

Where $\prod'(\boldsymbol{i} \in \boldsymbol{Local - World}) = \frac{\boldsymbol{M}}{\boldsymbol{m_0+t}}$

After t time steps, there will be a network with M = t+m_0 nodes and m $*$ t edges.

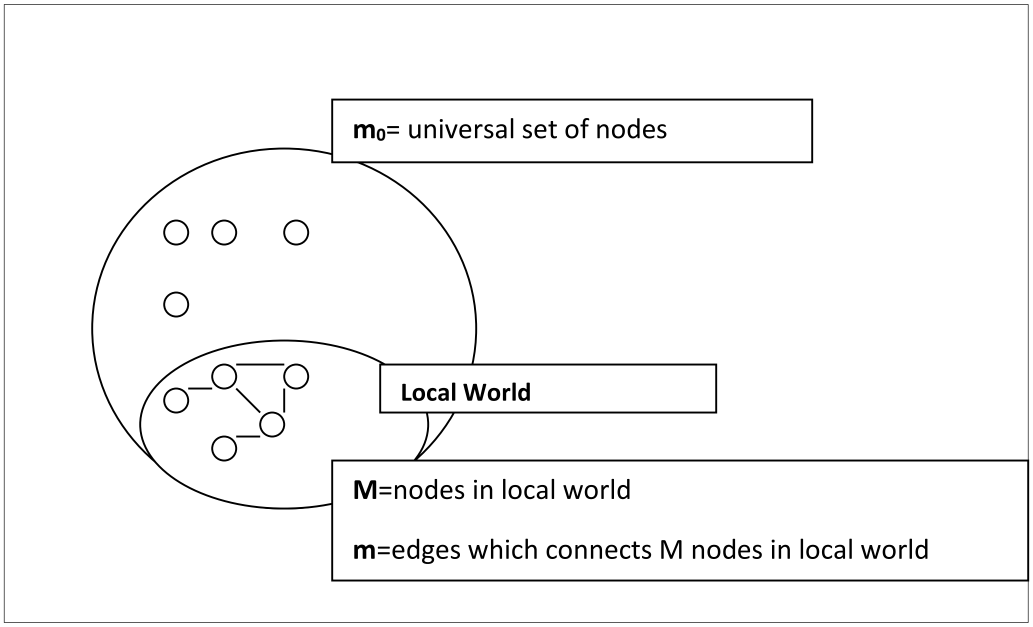


Figure 5.2: Illustration of various parameters in their roles as describing the local world and the universe

5.3. Applying Local World Model in WSN Topology Control Algorithms

To use Local World Model to capture functioning of WSN clustering algorithm, the model has to take into account two types of nodes: normal nodes and cluster nodes. The cluster nodes are the cluster heads, and the normal nodes are members in a cluster. There is only one cluster node attached to a normal node; in other words, the normal node has only one edge, which means that the normal node cannot relay data from other nodes. A cluster node can integrate and transmit data from other nodes. Both of these two types of nodes can connect to a cluster node, and the number of edges is limited in every cluster node because of its energy consideration. When a new cluster node joins the network, it is randomly assigned an initial energy $\mathbf{E_i}$ from the interval [$\mathbf{E_{min}}$, $\mathbf{E_{max}}$]. The limited number of edges in every cluster node is represented by $\mathbf{k_{max\_i}}$, which is based on the initial energy of the cluster nodes $\mathbf{E_i}$ where $\mathbf{k_{max}}$ [9] is given as follows.

$$\boldsymbol{k_{max\_i} = k_{max} * \frac{E_i}{E_{max}}}$$

$\mathbf{k_{max\_i}}$ reflects the ability to have the maximum number of edges for cluster node ***i***.

The growth model is described as follows. Starting with a small number of nodes, (all of them are cluster nodes), they randomly link each other. This results in an initial network.

1. **Growth**: At every time step, a new cluster node or a normal node with one edge enters into the existing network with a probability **p** or **1-p** respectively.If the new node is a cluster node, then it is assigned a random energy value of $\mathbf{E_i}$ as discussed. A small number of cluster nodes would cause many sensor nodes to link to them, which results in faster energy consumption; But a large number of cluster nodes

would be more wasteful regarding energy efficiency. Thus, the value of **p** is assumed to be in the range as ***0<p<0.5***.

2. **Preferential Attachment***:* A new node arriving at the network links to an old cluster node that is selected randomly from the already existing network. Nodes in WSNs have the constraint of energy and connectivity and only communicate data with the cluster nodes in their local area. First, **M** cluster nodes are selected randomly from the network as the new incoming node's local world; then, one of the cluster nodes is chosen to link with the new node according to the probability $\prod_{local}(k_i)$.

If the new incoming node is a cluster node, then the probability is set as follows.

$$\prod_{ki} = (1 - \frac{k_i}{k_{\max i}}) \frac{k_i}{\sum_{j \in local} k_i}. \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

In this case, when the value of **ki** is high, the probability that it will be chosen to connect to the new node is higher.

If the new incoming node is a normal node, then the probability is defined as follows:

$$\prod_{ci} = (1 - \frac{k_i}{k_{\max i}}) \frac{c_i}{\sum_{j \in local} c_i} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Where c_i is the number of edges of the cluster node *i*. The higher the value of c_i is, the higher the probability that it will be chosen to connect to the new node. Only through this approach we can adjust the number of cluster nodes that are linked to one cluster node (cluster head).

Total Preferential probability is given as follows.

$$\prod_{\boldsymbol{ki}} = \prod_{\boldsymbol{ki}} + \prod_{\boldsymbol{ci}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

In [9], authors have considered the expenditure of energy in the process of linking nodes together. The disadvantage is that the energy in a cluster node will be exhausted in only few rounds if self-organization is allowed. In fact, the energy consumption will be relatively low; if only **k** $_{\mathbf{max\_i}}$ is considered to be the limit for a cluster node to connect to others randomly.

Anti-Preferential Attachment: Let us consider a parameter **z** called the deletion rate or anti-preferential attachment factor, which is defined as the rate of links removed divided by the rate of connections added. It's observed that lesser the energy of the node, the more will be the probability of it being deleted. Let this probability be denoted as $\prod^{*}(k_i)$.

For the outgoing cluster nodes, we have

$$\prod^{*}(\boldsymbol{k_i}) \approx \frac{\mathbf{1}}{\boldsymbol{m_0} + \boldsymbol{p} * \boldsymbol{t}}$$

For the outgoing normal nodes, we have

.

$$\prod^{*}(\boldsymbol{c_i}) \approx \frac{\mathbf{1}}{\boldsymbol{m_0} + (\mathbf{1} - \boldsymbol{p}) * \boldsymbol{t}}$$

So the total anti-preferential probability is given as follows.

$$\prod^{*}(\boldsymbol{k_i}) + \prod^{*}(\boldsymbol{c_i}) = \prod_{\boldsymbol{ki}}^{*}$$

$$\prod_{ki}^{*} = \frac{\{\left(2*\frac{m_0}{t}\right)+1\}}{t*\left(\frac{m_0}{t}+p\right)*(\frac{m_0}{t}+1-p)}$$

The anti-preferential removal mechanism is more reasonable for deleting links that are anti-parallel with the preferential connection. It is also consistent with the functioning of clustering algorithms that runs in rounds in wireless sensor networks. The wireless nodes that do not have enough energy, that is the dead nodes, are to be removed from the system. Thus, anti-preferential removal phenomenon is reasonable for clustering algorithms.

Using mean field theory [144][83] a qualitative analysis of dynamic characterization of a wireless sensor network can be given. By the mean-field theory, the preferential and non-preferential attachment may be combined in the following differential equation.

$$\frac{\delta k_i}{\delta t} = M\prod_{ki}^{u} - M*z[\prod_{ki}^{*} + \sum_{j\in linked}\prod_{ki}^{*} k_j^{-1}\prod_{ki}^{*}]$$

From Li-Chen model we have,

$$\prod_{ki}^{u} = \prod(local\,world)\prod_{ki}$$

$$\prod(local\,world) = \frac{1}{m_0+pt}$$

For a single node in local world

$$\prod_{ki}^{u} = \frac{1}{m_0+pt}[\prod{}_{ki} + \prod{}_{ci}]$$

$$\frac{\delta k_i}{\delta t} = \frac{M}{m_0 + p*t} * \prod_{ki}^{*} - \frac{M*z}{m_0 + p*t}\left\lfloor \prod_{ki}^{*} + \sum_{j \in linked(i)} \prod_{kj}^{*} k_i^{-1} \quad \prod_{ki}^{*} \right\rfloor \text{----------------}$$

----- (4)

By mean field theory,

$$\sum_{j \,\in\, linked(i)} \prod_{\boldsymbol{k_j}}^{*} \boldsymbol{k_j^{-1}} \approx \mathbf{1}$$

Therefore the above equation can be rewritten as

$$\boldsymbol{\frac{\delta k_j}{\delta t} = \frac{M}{m_0 + p * t}} * \prod_{\boldsymbol{k}} - \boldsymbol{\frac{M * z}{m_0 + p * t}}[\prod_{\boldsymbol{k}}^{*} + \prod_{\boldsymbol{k}}^{*}]$$

Using equations 1, 2 , 3 we have,

$$\frac{\delta k_i}{\delta t} = \frac{M}{m_0 + p*t} * \left[p*(1-\frac{k_i}{k_{\max i}})*\bar{k}^{-1}{}_i + (1-p)*(1-\frac{k_i}{k_{\max i}})*\bar{c}_i^{-1} \right] - \frac{2*M*z}{m_0 + p*t}\left[\frac{2*m_0 + t}{(m_0 + p*t)*(m_0 + t - p*t)} \right]$$

..............................(5)

where

$$\frac{1}{\bar{k}_i} = \frac{k_i}{\sum_i k_i} \quad \text{and} \quad \frac{1}{\bar{c}_i} = \frac{c_i}{\sum_i c_i}$$

5.3.1 Analysis of the Dynamic Equation

CASE I:

If **z=0 , M=1** ie the new node selects node unless it reaches k. Moreover, the preferential attachment mechanism does not work. The rate of growth of $\mathbf{k_i}$ is as

$$\frac{\delta k_i}{\delta t} = \frac{1}{m_0 + p * t}$$

The denominator of the above expression is the number of cluster nodes at time **t.**

CASE II:

If M=m$_0$+p*t. This means that the local world is the whole network

$$\frac{\delta k}{\delta t} = p*(1-\frac{k_i}{k_{\max i}})*\overline{k^{-1}}_i + (1-p)*(1-\frac{k_i}{k_{\max i}})*\overline{c_i^{-1}} - \quad \frac{2*M*z}{m_0 + p*t}\left\lfloor \frac{2*m_0+t}{(m_0+p*t)*(m_0+t-p*t)} \right\rfloor$$

………………………..(6)

In a network, the degrees k$_i$ of most of the nodes are much smaller than their maximum k$_{max\_i}$; thus, we obtain the following formula.

$$1-\frac{k_i}{k_{\max i}} \approx 1 \quad \textbf{……………………………………………………..(7)}$$

Putting the value of equation 7 in equation 6 we have

$$\frac{\boldsymbol{\delta k_j}}{\boldsymbol{\delta t}} = p * \frac{\boldsymbol{k_i}}{\sum_{\boldsymbol{i}} \boldsymbol{k_i}} + (1-p) * \frac{\boldsymbol{c_i}}{\sum_{\boldsymbol{i}} \boldsymbol{c_i}} - \frac{2*M*z}{\boldsymbol{m_0}+\boldsymbol{p}*\boldsymbol{t}}\left[\frac{2*\boldsymbol{m_0}+t}{(\boldsymbol{m_0}+p*t)*(\boldsymbol{m_0}+t-p*t)}\right]\text{---(8)}$$

By definition,

$$\overline{k} = \frac{\sum'_{j} k_j}{k_i}$$

$$\overline{k} = total\_\deg ree\_of\_node\_universe / total\_\deg ree\_of\_nodes\_local - world$$

$$= \frac{m_0 + p*t + N}{m_0 + p*t} = \frac{m_0 + p*t + m_0 + t}{m_0 + p*t}$$

Therefore,

$$\bar{k} = \frac{2*m_0 + t + p*t}{m_0 + p*t} \quad \ldots\ldots\ldots\ldots..(9)$$

Similarly, we have for c_i

$$\overline{c} = \frac{\sum'_{j} c_j}{c_i} = \bar{k} - \frac{(1-p)*t}{m_0 + p*t} = 2 \quad \ldots\ldots\ldots\ldots\ldots\ldots(10)$$

(As the cluster node will have one such node attached to itself the status of that node is either another cluster head or normal head hence the count value of $\overline{c}$ is 2)

Equations (8) and (9) are used to find the values of $\overline{c}$ and $\bar{k}$.

Finally the following equation is formed after substituting the values of $\sum'_{j} c_j$ and $\sum'_{j} k_j$ in equation 8.

$$\frac{\delta k}{\delta t} = \frac{p*k_i}{2*m_0 + p*t + t} + (1-p)*\frac{c_i}{2*(m_0 + p*t)} - \frac{2*z*(2*m_0 + t)}{(m_0 + p*t)*(m_0 + t - p*t)}$$

At t-> infinity, m_0->0.

$$\frac{\delta k}{\delta t} = \frac{p*k_i}{p*t + t} + (1-p)*\frac{c_i}{2*p*t} - \frac{2*z*t}{p*t*(t - p*t)}$$

Reducing the above equation by assuming **A=p/1+p B=1-p/p**, **C=1/p*(p-1**) and **c_i=2**

$$\frac{\delta k}{\delta t}=\frac{A*k_i}{t}+\frac{B}{t}-\frac{2*z*C}{t} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(10)$$

With initial conditions are given as **$k_i(t_i) = 1$**.

By integration, we have the solution as

$$\frac{t}{t_i}=\left(\frac{k(t)*A+B-2*z*C}{A+B-2*z*C}\right)^{1/A}$$

Moreover, to find the degree distribution **P(k)** i.e., the probability that a node has **k** edges), we first calculate the cumulative probability **P[k_i(t) < k]** . Suppose that the node enters into the network at equal time intervals. We define the probability density **t_i** of as follows.

$$P(t)=\frac{1}{m_0+t}$$

So **P[k_i(t) < k]**has the following form

$$(1-\frac{t}{t+\boldsymbol{m_0}}\left(\frac{k*A+B-2*z*C}{A+B-2*z*C}\right)^{\frac{1}{A}})$$

Hence, the degree distribution P(k) can be obtained:

$$\frac{\partial}{\partial t}(1-\frac{t}{t+\boldsymbol{m_0}}\left(\frac{k*A+B-2*z*C}{A+B-2*z*C}\right)^{\frac{1}{A}})$$

$$=\frac{t}{(t+\boldsymbol{m_0})}(A+B-2C)^{\frac{1}{A}}(A*k+B-2*z*C)^{-1-1/A}$$

$$\approx (A+B-2*C)^{1/A} * (k*A+B-2*z*C)^{\frac{-1}{A}-1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(11)$$

Equation 11 denotes the degree distribution function to understand the dynamics of clustering algorithms. Putting back the values of A, B, C we have the distribution function asthe degree distribution P(k) the probability that a node has k edges is

$$P(k) \approx \left(\frac{p}{p-1} + \frac{1-p}{p} - \frac{2*z}{p*(1-p)} \right)^{\frac{1+p}{p}} * \left(k*\frac{p}{1+p} + \frac{1-p}{p} - 2*\frac{z}{p*(1-p)} \right)^{\frac{-2*p-1}{p}}$$

$$\ldots\ldots\ldots\ldots(12)$$

Next, the value of **z** is computed that maximizes Equation 12.

$$\frac{dP(k)}{dz} = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (13)$$

By solving Equation (13) we obtain the value of **z** as

$$z = \frac{(1+A+B-k)}{2*C} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(14)$$

Considering the anti-preferential factor (z) to be minimum, during the evolution of the network as the result the value of **z** will tend to zero.

By setting the value of **z** to zero, we have the following relation

$$p = \frac{1}{2}\left\lfloor \left(\frac{k+3}{k-1} \right)^{0.5} - 1 \right\rfloor \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (15)$$

Equation 15 states the probability of clustering when the antiprferential attachment factor is z is zero.

5.4. Analysis of Toplogy Control Algorithms

In this section, we have carried out a simulation of three localized topology control protocols, namely, A3, Simple tree Protocol, CDS Rule K. Simulation was carried using the Atarraya [16] Simulator. A total number of nodes was 100, 200, 300, 500 respectively and they were tested for A3, Simple Tree Protocol and CDS Rule K. The output was recorded for the average degree of nodes, **k,** for three protocols mentioned earlier.

The following diagram, Figure 5.3, shows an example of clustering and computation of average **k** value.

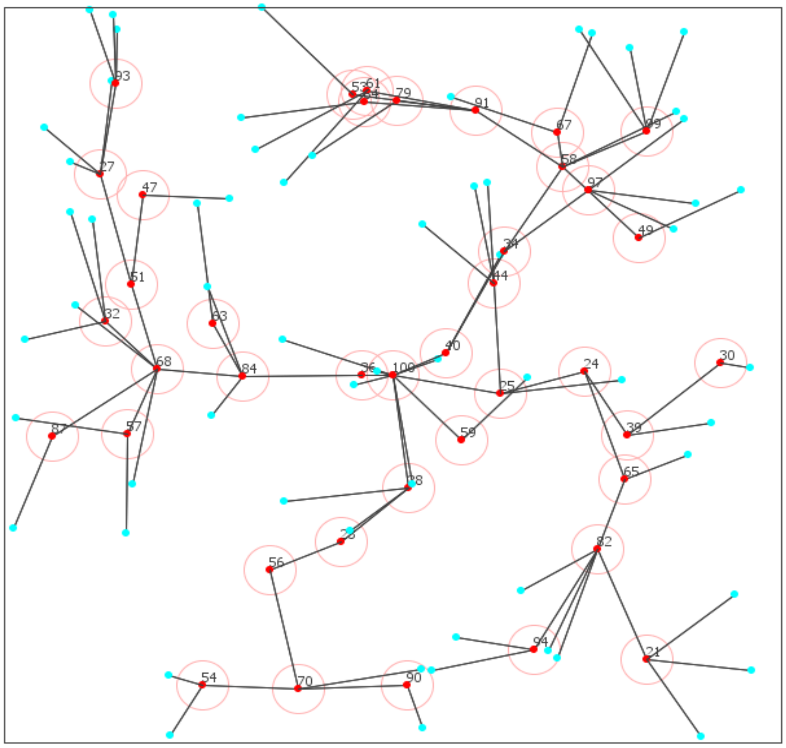


Figure 5.3 Clustering as obtained by running simple tree protocol in a network of 100 nodes

This above figure shows the cluster heads in red circles with which normal nodes are attached. Taking the number of all the neighboring nodes of every cluster heads and dividing by the number of cluster heads we obtain the average value of **k.** By similar procedure we have obtained the average value of **k** for a different number of nodes.

We have also carried out simulation study using Simple Tree Protocol for a different number of nodes. The number of clusters obtained using theoretical calculation (Equation 15) is compared against the number of clusters obtained using simulation study, which is presented in Table 5.1.

N	**k**	**p**	**n(T)=p*N**	**n(E)**
100	3	0.366	37	40
200	3	0.366	73	71
300	4	0.263	79	71
400	4	0.263	105	96
500	4	0.263	132	130

Table – 5.1: N= Number of nodes, k=no. of neighbors, p=probability of selecting a cluster head (ref. equation 15), n(T)=Theoretically calculated number of clusters, n(E)=Experimentally calculated number of clusters

Similar simulation study was also carried out using CDS-Rule k Protocol for different number of nodes. The number of clusters obtained using theoretical calculation (Equation 15) is compared against the number of clusters obtained using simulation study, which is presented in Table 5.2.

N	**k**	**P**	**n(T)=p*N**	**n(E)**
100	4	0.264	27	27
200	6	0.171	35	38
300	10	0.101	31	33

400	10	0.101	41	43
500	13	0.077	39	43

Table – 5.2: N=Number of nodes, k=no. of neighbors, p=probability of selecting a cluster head (ref. equation 15), n(T)=Theoretically calculated number of clusters, n(E)=Experimentally calculated number of clusters

Similar simulation study was also carried out using A3 Protocol for different number of nodes. The number of clusters obtained using theoretical calculation (Equation 15) is compared against the number of clusters obtained using simulation study, which is presented in Table 5.3.

N	**k**	**P**	**n(T)=p*N**	**n(E)**
100	4	0.263	27	34
200	6	0.171	35	40
300	8	0.126	38	49
400	10	0.101	41	48
500	12	0.083	42	46

Table 5.3: N=Number of nodes, k=no. of neighbors, p=probability of selecting a cluster head (ref. equation 15), n(T)=Theoretically calculated number of clusters, n(E)=Experimentally calculated number of clusters

5.4.1. Discussion on A3 and Simple Tree Protocol

Next, we plot the theoretically calculated number of clusters, and experimentally calculated number of clusters as shown in Table 5.1 and 5.3 for A3 and Simple Tree Protocols.

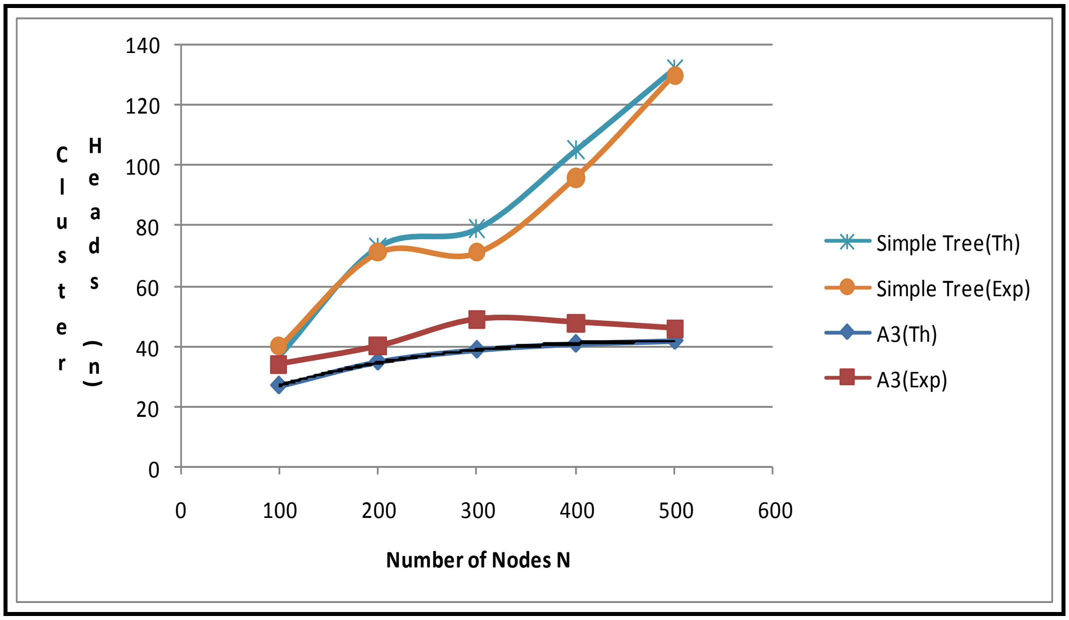


Figure 5.4: Number of Nodes vs Number of cluster heads in Simple Tree protocol and A3 Protocol

We observe the following.

1. In A3 and Simple Tree protocols, the curves for a theoretically calculated number of clusters and experimentally calculated number of clusters follows fourth-degree equation. The trend line polynomial options of MS Excel have been used to study the degree of curve fit of the curves to obtain the following polynomials.

   $n = -4E\text{-}10N^4 + 8E\text{-}07N^3 - 0.000N^2 + 0.197N + 12$ (A3 protocol)

   $n = -3E\text{-}08N^4 + 4E\text{-}05N^3 - 0.016N^2 + 3.164N - 148$ (Simple tree protocol)

2. In Figure 5.4, there is gap between the theoretical and experimental curves in case of A3 and Simple Tree protocol. This may be attributed due to the fact that A3 (approximate CDS) only preserves 1-connectivity whereas the Simple Tree protocol has multiple connectivity. So, A3 protocol requires less energy to construct the tree as compared to spanning tree which confirms to the experimental results

5.4.2. Discussion on CDS Rule K

Consider Equation (14) which is mentioned in the following.

$z = \frac{(1 + A + B - k)}{2 * C}$ Where A, B, C, k, z have been defined previously.

Table 5.4 enumerates the value of z for various values of k corresponding to a different number of nodes in the network.

Number of Nodes (N)	**k**	**p**	**z**
100	4	0.5	-7
200	6	0.5	-9
300	8	0.5	-11
400	10	0.5	-13
500	13	0.5	-16

Table-5.4: Value of z for different k values in networks of dufferent sizes

Combining the results of Table 5.2 and Table 5.4 we obtain the following graph as

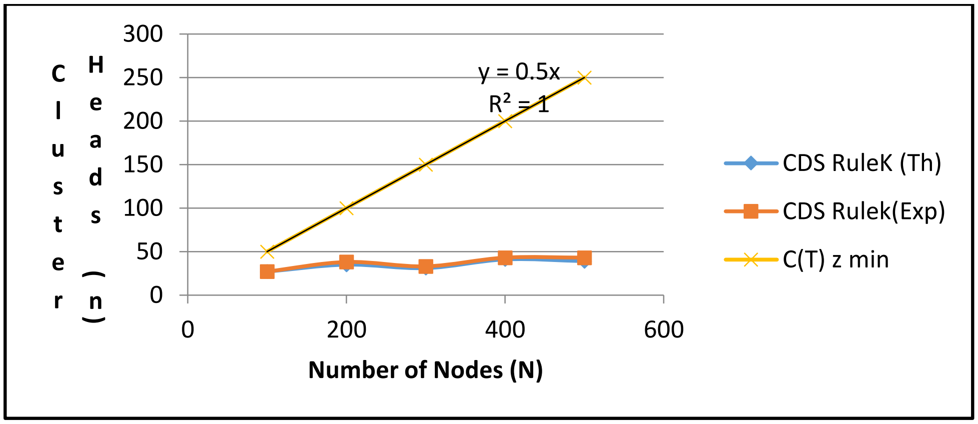


Figure 5.5: Total Number of nodes(N) vs Number of Cluster Heads in A3 Algorithm

In Table 5.4, the minimum value of z corresponds to a probability value 0.5 which indicates that the clusters have dissociated into one cluster head and one normal node which ideally predicts the case when the energy of the nodes have drained out.

In Figure 5.5, it can be seen that the mechanics of clustering in CDS- Rule K is in accordance with the assumptions considered while deducing the distribution function using mean field theory. Unlike A3 and Simple tree protocol, CDS-Rule K also exhibits a relation of degree 4 as shown ($n = -2E\text{-}08N^4 + 3E\text{-}05N^3 - 0.012N^2 + 2.085N - 87$). The linear graph as shown in the figure represents the situation when the value of **z** is min. This graph is linear due to a constant value of **p** (0.5 here) where as other graphs were drawn on constant **z** (0 here).

Above experimental results lead us to the following explanation of the anti-preferential factor **z.**

The anti-preferential factor can have three distinct values

a. When z = 0 then the effective degree of nodes is equal to the analytically obtained degree of a node. Here the number of cluster heads formed will be optimal.

b. When z > 0 then a node will have fewer connections available. Here the number of cluster heads will be higher.

c. When z < 0 then a node will have more connectivity. Here the number of cluster heads will be lowest as a result will lead to the association.

5.5. Conclusions

In this chapter, we have tried to provide a framework to model tree based clustering algorithms in a WSN formally. Based on the formalism, we have theoretically calculated some parameters such as the number of cluster heads, average number of degree for a given algorithm. The theoretical results tally with results obtained by simulation studies. We have introduced a factor called *z* in network evolution model. It seems that the *z* factor has an impact on the functioning of these protocols. We keep that study as a future exercise. The research at the current stage is more suitable for modeling tree-based clustering algorithms that work on the top of Connected Dominating Set construction. Algorithms that are based on more parameters have to be modeled appropriately in the framework before application.

In recent papers energy distribution has been considered in network evolution model for Wireless Sensor Netwroks. But, so far, the framework of network evolution model has not been used to capture the characteristics of clustering algorithms.

Chapter 6
Performance Analysis of Some Cluster based Topology Management Protocols

In distributed computing, clustering of the nodes is generally used to make the communication process energy-efficient. However, in the mechanics of clustering, the number of clusters increases as the energy of the nodes gets depleted. This dispersive nature of clustering probability leads to the quick death of the nodes. This chapter explains the usage of an optimization matrix from clustering probability as obtained from a network evolution model. The proposed framework of an optimization matrix shows considerable promise in boosting the efficiency of data delivery and network lifetime of the hierarchical clustering protocols in wireless sensor networks.

6.1. Introduction

Real large-scale networks – whether biological, social or communication – are complex dynamical systems. Studying the properties of these networks allows us to control and predict the behaviour of such systems [L8]. The networks mentioned above can be roughly split into two types: 1) small-scale networks and 2) large-scale networks. In understanding large-scale network dynamics, concepts such as preferential attachment and anti-preferential attachment, and a mainly probabilistic approach have been used. However, in the understanding of small-scale networks, the graph-theoretic approach is enriched with the concepts of soft computing and probability. The authors in [30] have reasonably succeeded in approximating the usage of large-scale networks into small-scale networks of the wireless sensor networks (WSN) domain.

The latest research [148] shows, by combining the concept of communication energy principles and the geometry of the field, that clustering is independent of network size, and the energy consumed by the transmitter circuitry has no impact on the optimal cluster size. Moreover, receiver circuitry can influence the clustering that occurs.

In the existing literature [102, 108] there is evidence of designing newer protocols with the scale-free concept, although there is hardly any analysis on the improvement or review of the current protocols with complex network theories [134].

Amongst WSN protocols, the LEACH protocol [158, 166] is one of the hierarchical clustering routing protocols in wireless sensor networks which uses a probability model in the selection of the nodes to be cluster heads or common nodes. This probabilistic approach makes LEACH attractive as a system to experiment on using other probability-based models.

Although there are many variants of LEACH protocols [50, 104, 132, 183], we found that LEACH has a reputation for being versatile and is used in numerous real-life applications [28]. This makes it suitable for our endeavour to experiment on a Network Evolution Model with LEACH.

The remainder of this chapter is organized into the following sections: section 6.2 gives a brief introduction to LEACH and the network evolution model, section 6.3 discusses the problem statement and the proposed framework, section 6.4 gives details on the setup of the experiment, section 6.5 provides a discussion, and finally section 6.6 explains the conclusions and future work.

6.2. Related Reviews:

6.2.1. LEACH Protocol

LEACH (Low Energy Adaptive Clustering Hierarchy) [158] is a protocol for micro wireless sensor networks that achieves low energy loss with high-quality application specific delivery. In this architecture, the nodes collaborate locally to reduce the amount of data to be sent to the end user. It has been found that the proximity of the nodes actively allows the data to be correlated. Therefore, clustering architecture is used for data dissemination in the LEACH protocol. This design uses a node designated to be a cluster head to receive data from the other sensor nodes. This cluster head takes the responsibility of reducing the received data signals into actual data while maintaining the useful information content. As there is no fixed infrastructure to collect the data, the cluster head must be rotated around other members to increase the lifetime of the network. For the rotation of cluster heads, the cluster forming algorithm should ensure minimum overheads in terms of energy and time.

Despite all the advantages of the LEACH protocol, it suffers from producing quality clusters by dispersing the cluster heads throughout the network. Therefore, LEACH-C (LEACH-Centralized) is a protocol that uses the central base station for computing the best cluster heads with an expensive algorithm such as simulated annealing or self-organizing maps coupled with K-means [66]. These are being used to determine optimum clusters as this is an NP-hard problem.

To obtain a good cluster, it is fundamental that the base station ensures that energy is evenly distributed throughout all the nodes. To achieve this, the base station computes the average energy of all the nodes and separates the nodes with low energy from the nodes with average energy. These segregated nodes with energy higher than the average level run the cluster selection algorithm, thereby reducing the computation overhead.

6.2.2. Performance Areas in WSNs Due to Scale-Free Networks

1. **Survivability**: The applications of WSNs range from important societal issues such as environmental and habitat monitoring to economic issues such as production control and structure monitoring [101, 192, 152, 16]. WSNs are vulnerable to energy depletion due to battery drain or an internal problem in the node. In addition, natural disasters and deliberate attacks can lead to WSNs being collapsed easily. Therefore, improving the survivability of the network is an important concern in the study of WSNs [16]. In [189], the authors presented two scale-free topology evolution models based on complex network theory. The residual energy, node fitness, node saturation and node communication range were taken into account during the topology evolution. These models achieved a reasonable distribution of cluster heads in the topology via scale-free random walks, thereby ensuring the survivability of the network.
2. **Fault Tolerance and Intrusion Tolerance**: [98] The authors used the mathematical expression of topological degree distribution to analyse its effect on the properties of topological fault-tolerance and topological

intrusion-tolerance. The optimal scale-free topology for minimizing the fault and intrusion was obtained. It could be observed that the scale-free topology was more robust against random faults. Moreover, it somewhat mitigated the chance of selective remove attacks on the network and enhanced its lifetime. In [193] the authors proposed a distributed algorithm, allowing each node to control its transmission power and choose its neighbour based on certain weights, which are real random numbers following the negative power-law probability distribution. This distributed algorithm seems to be scale-free, strongly connected and bi-directional.

3. Localization of Nodes in Sensor Networks: WSNs are normally deployed in hostile conditions where human interference is practically impossible. Physically, these areas can be referred to as non-uniform anisotropic networks with holes. However, this kind of terrain results in a low accuracy of node localization due to the existence of holes and the effect of the Euclidean distances between nodes. Therefore, in [160] the authors proposed a Heuristic Multidimensional Scaling (HMDS) algorithm to improve the accuracy of node localization in anisotropic WSNs with holes. In this algorithm, which uses the concept of virtual nodes and constructs the shortest possible paths between nodes, the Euclidean distances between nodes are obtained by employing a heuristic approach. The HMDS algorithm greatly reduces the communication complexity and computational complexity compared to the MDS-MAP algorithm. Simulation results demonstrate that the HMDS algorithm requires a smaller number of virtual nodes to obtain the node locations. The HMDS algorithm is suitable for four different topologies, including the semi-C-shape topology, the O-shape topology, the multiple O-shape topology and the concave-shape topology, and is highly successful in determining the Euclidian distances between nodes. In [186], the authors studied the problem of localizing a large sensor network with a complex shape, possibly with holes. A major challenge concerning such networks is to establish the correct network layout, for example avoiding global flips in which a part of the network folds on top of another. The implementation of the algorithm firstly selects landmarks on network boundaries with sufficient density, then constructs the landmark Voronoi diagram and its dual combinatorial Delaunay complex on these landmarks. By simply gluing the Delaunay triangles onto nicely localized spaces, the rest of the nodes can easily localize themselves by trilateration to nearby landmark nodes. Therefore, a practical and accurate localization algorithm for large networks using only network connectivity can be achieved.

4. I**mproved Stability of the Network**: In [49, 187], the authors designed an epidemic model for WSNs based on limited scale-free networks. This implementation led to a positive equilibrium being determined, with some restrictions, which in turn proves the stability of the system. The restriction on the WSN system was done by partitioning the network into higher-degree motes and lower-degree motes and then equating the degrees of all higher-degree motes with lower-degree motes, yielding a restricted scale-free network. The unique portioning of this network reduced the complexity of the network under consideration.

5. **Optimal Deployment:** Questions such as how many high-end sensors to be used and whether to deploy them in an ordered or random manner require proper consideration. In [133], the authors describe a novel scheme representing a wide variety of scenarios ranging from totally random to planned stochastic node deployment, in both homogeneous and heterogeneous sensor networks. Using only around 3% of the high-end sensors and deploying nodes by using this slightly attractive model, improved characteristics of the network topology can be observed, such as (i) a low average path length, (ii) a high clustering coefficient, and (iii) an improved relay task distribution between sensors. Moreover, the authors also provide a model that can improve the network lifetime and diminish the energy hole effect.

6. **Energy Efficiency:** Based on complex network theory, the authors in [193] presented two self-organized energy-efficient models for wireless sensor networks in their paper. The first model constructed wireless sensor networks

according to the connectivity and remaining energy of each sensor node. In the second model, not only the remaining energy was considered, but the constraint of links to each node was also introduced. This could therefore produce scale-free networks with a performance of random error tolerance. This model could make the energy consumption of the entire network more balanced. Finally, the numerical experiments of the two models was presented.

7. **Performance Enhancement of Existing Algorithms**: The authors in [186] show that scale-free topologies have a positive impact on the performance of gossiping algorithms in peer-to-peer overlay networks. This result is important in the context of ad hoc networks, where each node participates in controlling the network topology. Therefore, by using the scale-free concept, when combined with such topologies, typical gossiping algorithms tend to require fewer messages and experience smaller latency than when combined with other topologies, such as rings or grids. This suggests that if the topology control scheme is aimed at producing scale-free characteristics in an overlay network, then the performance of these protocols seems to be better.

9. **Mobility Models**: The author in [141] proposed the clustered mobility model (CMM), which facilitates the formation of hubs in a network, satisfying the scale-free property. With the CMM, it is possible to control the degree of node concentration or non-homogeneity to easily assess the strengths and weaknesses of the scale-free phenomena. The most important feature of the CMM is that it does not possess any unintended spatial or temporal characteristics found in other mobility models.

6.2.3. Network Evolution Model

In [30], the authors analytically used complex network theory to quantify some of the observed properties of topology control algorithms. To build this framework, the probabilistic approach of the Li-Chen model [185], the local world model, was used to mimic the wireless sensor network. In addition, the dynamics of clustering were addressed by the concept of preferential and anti-preferential attachment. The anti-preferential removal mechanism is more reasonable for deleting links which are anti-parallel with the preferential connection [30, 76]. It is also consistent with the functioning of clustering algorithms that run in rounds in wireless sensor networks. The wireless nodes that do not have enough energy, also known as the dead nodes, must be removed from the system. Therefore, the anti-preferential removal phenomenon is reasonable for clustering algorithms. Finally, combining the mathematical realizations of these facts with mean field theory, we can obtain the distribution function as the degree distribution P(k) , where P(k) is the probability that the node has k edges. This distribution is further minimized concerning the anti-preferential attachment, which during the evolution process tends to zero as this phenomenon of non-attachment to a preferential neighbour is absent for wireless sensor networks. This considerably reduces the distribution function to yield.

$$p = 0.5 * \left\lfloor \sqrt{\frac{k+3}{k-1}} - 1 \right\rfloor \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

The above expression is known as the probability of clustering in the network. To draw a meaningful graph of the above equation to express the usefulness of the wireless sensor network domain, we can convert the p vs. k to the number of clusters vs. the number of connections, by multiplying the number of nodes in the network (100 nodes in this case) to the probability obtained for different connections (k). Figure 6.1 shows the nature of the graph.

Problem: Show the validity of a random 100-node WSN network simulating a 5-cluster head network.

Solution: the average number of connections in a cluster of a 5-cluster 100-node network = (100/5) -1=19 connections.

To check the validity, we therefore put the value of k in equation (1) and obtain the value of P as 0.0527707984. Multiplying back with 100 we obtain

5.27, approximately 5 (considering the floor value w.r.t 5.5). This is the number of clusters possible with a 100-node network.

For a 6-cluster 100-node network, the clustering probability value is 0.0627314339. Similarly, following the previous caveat, the number of clusters will be 6. Similarly, for a 4-cluster 100-node network, the theoretical number of clusters is 4.15 which can be rounded off to 4. For a 3-cluster 100-node network, the theoretical number of 3.03 is rounded off to 3, and the same pattern is followed for 2 and 1. However, for a 7-cluster 100-node network, the clustering probability value is 0.0773502692, so the number of clusters will be 8. Therefore, there is disparity between the desired number of cluster heads to the number calculated (tabular explanation in Appendix C). This indicates a deviation from the optimized plane.

Therefore, good clusters can range from 1% to 6% of the total nodes. This result confirms the mathematical deduction of Heinzelman et al. which states that the clustering percentage (k_{opt}) lies between 1% and 6% of the nodes as cluster heads for the LEACH experiment [158]. With this important success, we can now proceed with the application of the equation in modifying the LEACH-C protocol for enhancing the energy efficiency, throughput, and existing nodes.

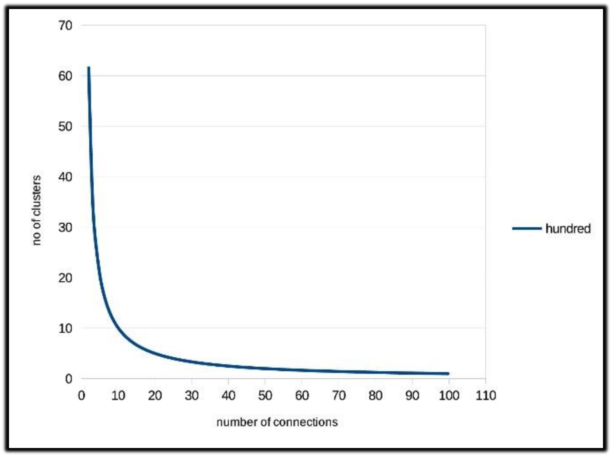


Figure 6.1: Plot showing the number of clusters to number of connections when N=100

6.3. Problem Statement and Proposed Framework

6. 3.1 Problem Statement

Consider the simulations of LEACH-C in NS-2 of a 5-cluster 100-node random network with the initial simulation parameters as energy per node 2 Joules. The output of the simulation is alive 4, data transmitted 67800, energy expense 198.160, and the number of rounds 510. These simulations have been done keeping the cluster heads fixed at 5 by assumption [158, 148].

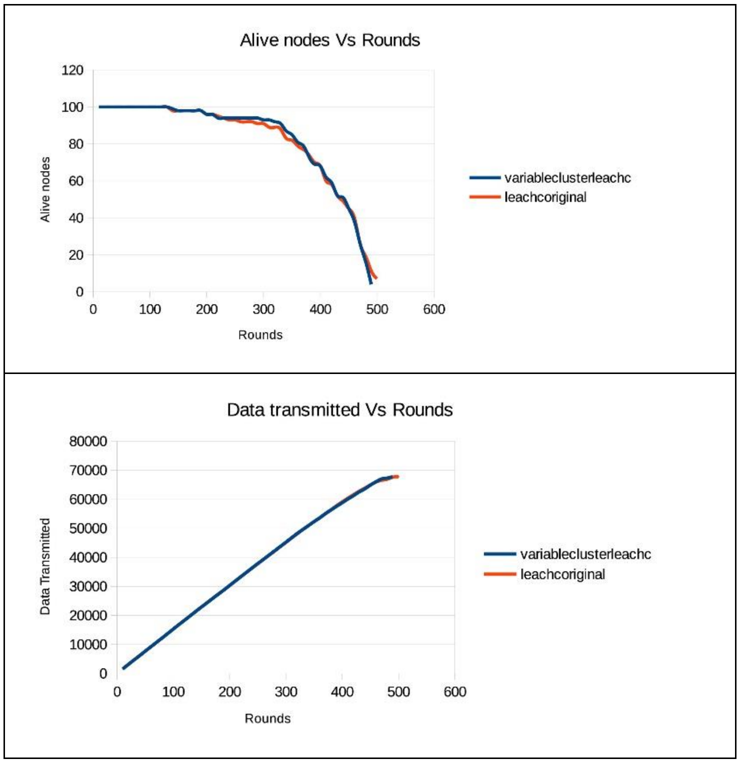
On the contrary, we find that as the simulation proceeds, the number of existing nodes decreases, therefore variable clustering should be evident to maintain the balance between cluster heads and non-cluster heads. For implementation, we took the reference of equation (1) regarding wireless sensor networks and used it to determine the variable clustering throughout the rounds. The details of the implementation algorithm are described in Scheme-I of Appendix B. The output of the performance is shown in Table 6.1. We can observe that there is almost no variation in the values when compared to the parameters. Instead, it seems that varying the clusters leads to lowering the output characteristics values. The detailed table is shown below.

Parameters	**Original LEACH-C**	**Variable Clustering (diverging) LEACH-C**
Alive	4	4
Data Transmitted	67800	67760
Energy loss	198.16	199.3258
Rounds	510	490

Table 6.1: Comparison between original LEACH-C and variable clustering LEACH-C

To investigate the deteriorating performance more closely, we would use an optimizing table (Appendix A) to visualize the original LEACH-C and the variable clustering LEACH-C. The optimizing table has existing nodes as a column header and the probability due to connections available as a row header. The elements of this matrix constitute the cluster heads for any particular node.

Therefore, for the variable clustering LEACH-C, the yellow boxes are plotted as the clusters that were made available at various times for the corresponding live nodes. Similarly, for the original LEACH-C where the cluster number was fixed (at five in this case), the number of clusters in the optimizing table is plotted as cyan. It is evident that initially both algorithms were performing the same, although after 380 rounds the variable clustering LEACH-C suffered casualties which it could not recover and finally ended at 490 rounds, whereas the original LEACH-C carried the simulation up to 510 rounds.



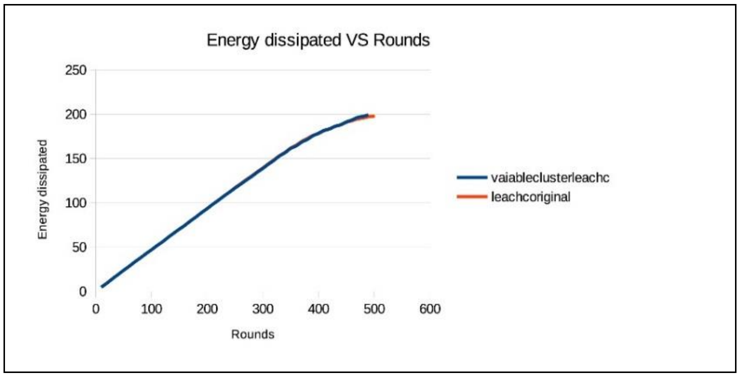


Figure 6.2: Plot showing alive, data transmitted, and energy spent versus rounds (200 nodes with 2J each node). [Original Leach-C to variable Leach-C]

So far, we have discussed the use of an optimizing table. The next section will address the main question of whether we can improve the output characteristic of LEACH-C. It will discuss our improvement using an optimization matrix.

6.3.2 Proposed Solution

The solution to this problem is more physical than algorithmic. Careful observation of the optimization table depicts that movement of the yellow boxes and the cyan boxes occurred in a diagonal direction (Appendix-A). This implies an increase in the number of cluster heads for the variable clustering LEACH-C and that the number is fixed in the case of the original LEACH. Another important aspect is that the cluster heads either increased or remained fixed for the decreasing number of live nodes which shows that an increase in the number of cluster heads leads to the dissipation of more energy. Using this observation, we can logically vary our cluster row-wise as shown by the green boxes, beginning from the starting point of the variable clustering LEACH-C and the original LEACH-C. From the optimization table we can observe that as the alive nodes reduce, the number of cluster heads also reduces.

As the optimization table does not say anything about the energy of the system under consideration, it is difficult to say whether the green path will be energy efficient or not, but it is at least the shortest path in terms of the alive nodes' connection space.

6.4. Simulation Details

The simulation of the original LEACH-C, the variable clustering LEACH-C and the modified variable clustering LEACH-C were simulated in NS-2.34. We installed NS-2.34 on an Intel P-IV, 512 MB RAM 32-bit machine using Ubuntu-14.01. This simulator was chosen because it has a proper radio model and a mac protocol, and complex scenarios can be easily tested. For our experiments, we used a 100-node network where nodes were randomly distributed with the BS at the location at the origin. The bandwidth of the channel was set to 1 Mb/s, each data message was 500 bytes long, and the packet header for each type of packet was 25 bytes long. The following figures describe the output of the experiment.

Parameters	Original LEACH-C	Modified LEACH-C
Alive	9	10
Data Transmitted	56886	61190
Energy loss	396.613	393.6059
Rounds	444	720

Table 6.2: A comparison of performance between the original LEACH-C and the LEACH-C Modified for 200 nodes, 2J each node

Parameters	Original LEACH-C	Modified LEACH-C
Alive	4	5
Data Transmitted	679709	693918
Energy loss	1978.506	1970.248
Rounds	4850	6000

Table 6.3: A comparison of the performance between the original LEACH-C and the LEACH-C Modified for 100 nodes, 20J each node

We assumed a simple model for the radio hardware energy dissipation, in which the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver consumes energy to run the radio electronics.

For the experiments described here, both the free space (power loss) and the multipath fading (power loss) channel models were used, depending on the distance between the transmitter and receiver. For the experiments described in this chapter, the communication energy Eelec =50 nJ /bit,ef=10 PJ bit m, parameters were set as emp=0.0013pJ bit m, and the energy for data aggregation was set as 5 nJ bit signal [1]. The details of the algorithm for the LEACH-C Modified (Scheme II) and the original LEACH-C are explained in Appendix B.

After observing the output characteristics, we found that there was some improvement in performance regarding alive nodes, data transmitted and energy dissipated. Table 6.1 below shows the vis-a-vis performance. Graphically and numerically examining the data, we found that the work of the optimization matrix was practical after the death of some nodes (the 73rd node in the case of LEACH-C Modified). Instead of stretching the reduced number of nodes (alive nodes) to more clusters, we had reduced the number of clusters.

We could roughly say that this procedure was effective with 27 percent of the nodes in the last cycle. We then ran the experiments with the nodes with higher energy to capture the manifestations at the end of the period.

Keeping the above observation in mind, we designed another two experiments. The first simulation had 100 nodes with 20 joules of energy for each node, and the second simulation had 200 nodes with 2 joules for each node. The purpose of these operations was to capture the management of clustering at the end of the cycle.

Figures 6.3, 6.4 and Tables 6.2, 6.3 describe the output graphs of the experiment mentioned above. The analysis of the operation with 200 nodes at 2J per node is worth noting, as it explains the effect of the modified LEACH-C protocol on clustering at the end cycle of the test. The effect is more pronounced when there is a large number of nodes, rather than when there are fewer nodes with higher energy.

LEACH was also considered for simulation purposes. The outcome of the simulations is shown in Figures 6.5, 6.6 and Tables 6.4, 6.5 respectively, for two different cases.

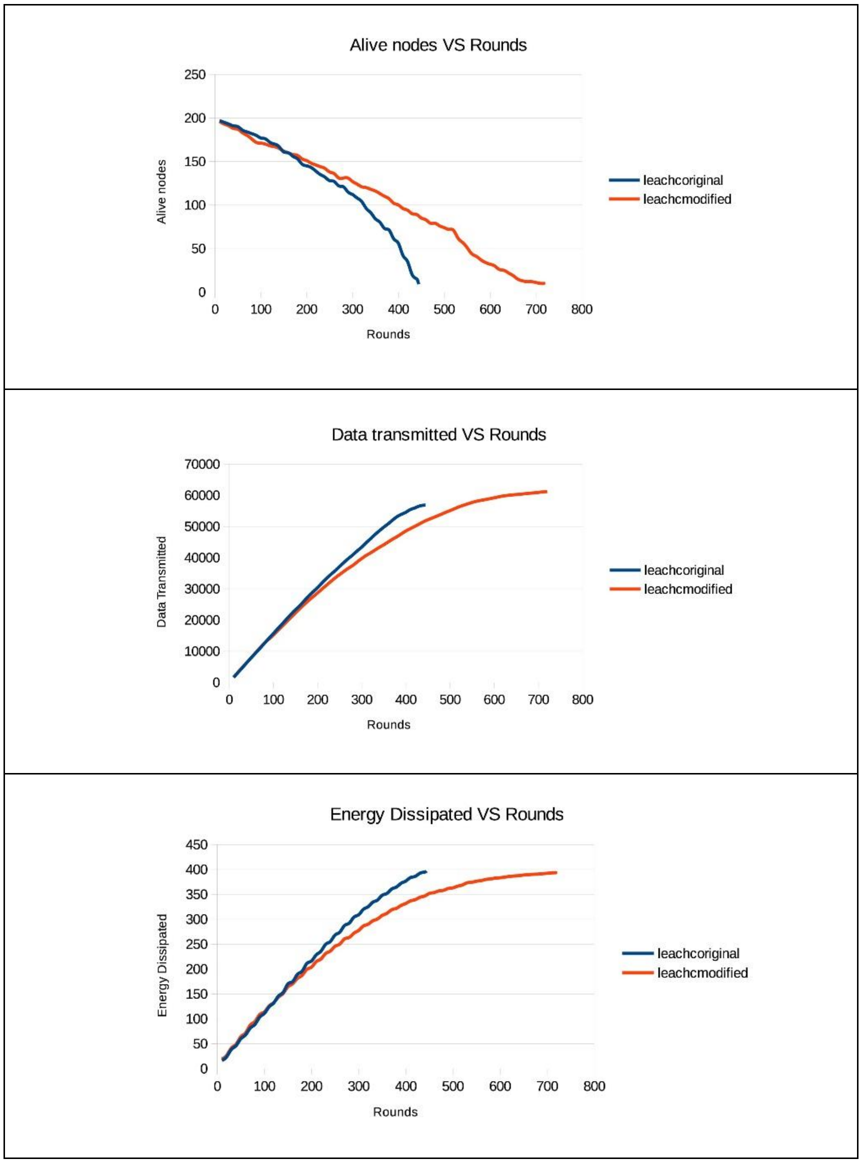


Figure 6.3: The plot of alive , data transmitted, and energy spent versus rounds (100 nodes with 20J each node) [original LEACH-C to LEACH-C modified]

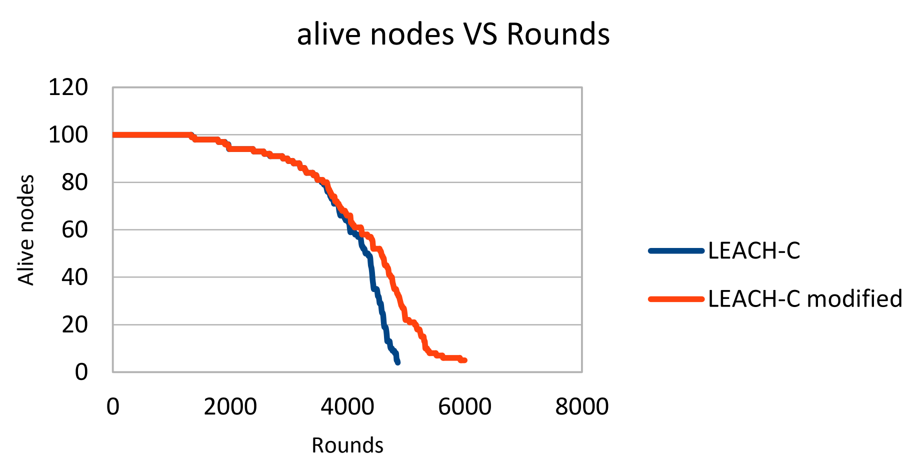
Parameters	**Original LEACH**	**LEACH modified**
Alive	4	4
Data Transmitted	519831	543903
Energy loss	1985.088	1997.186

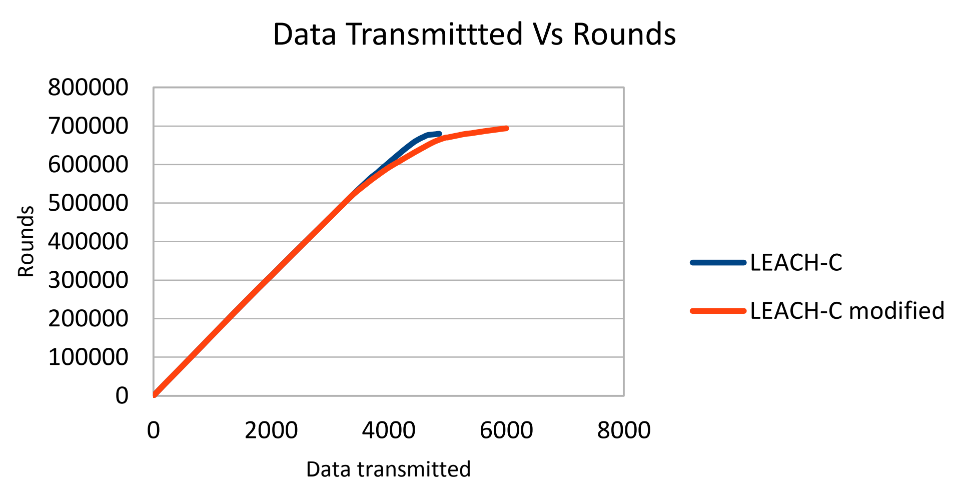
Rounds	5470	8870

Table 6.4: A comparison between the performance of original LEACH and LEACH Modified for 100 nodes, 20J each node

Parameters	**Original LEACH**	**LEACH modified**
Alive	9	9
Data Transmitted	18130	31274
Energy loss	395.136	397.397
Rounds	280	780

Table 6.5: A comparison between the performance of original LEACH and LEACH Modified for 200 nodes, 2J each node





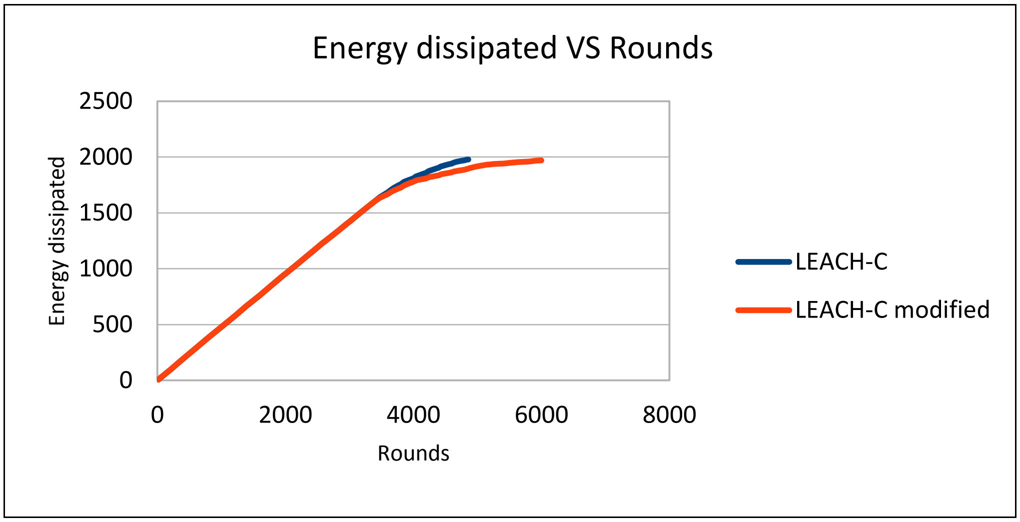
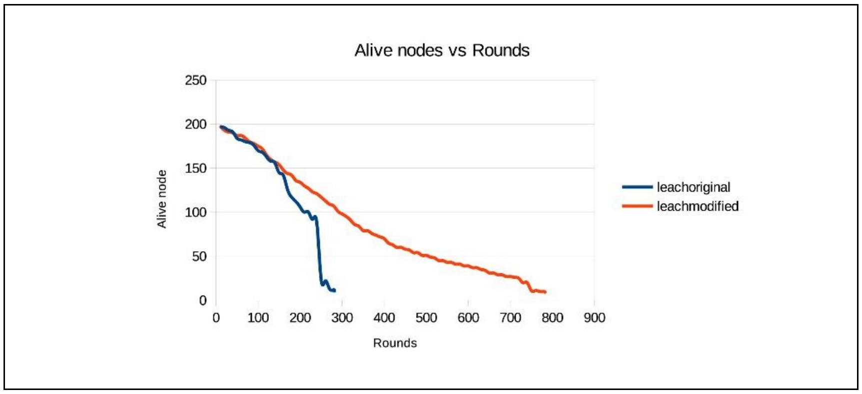


Figure 6.4: Plot showing alive, data transmitted, and energy spent versus rounds (100 nodes with 20J each node). [original LEACH-C to LEACH-C modified]

Close examination of the data reveals that after the death of 50 percent of the nodes, the modified LEACH performs very well in terms of the nodes' energy and life expectancy. However, this is even more evident in the case of 200 nodes. It is worth noticing that the energy expense graph of the original LEACH as described in Figure 6.6 states how abruptly the energy is used up by the remaining 50 percent of the nodes. This unexpected energy expense is due to the unstable clustering of the nodes, which can be seen as mitigated by the smooth graph of the modified LEACH in Figure 6.6. Therefore we can infer that proper management of clustering is acheieved at the end of the cycle which is reflected by an increased lifetime of the nodes.



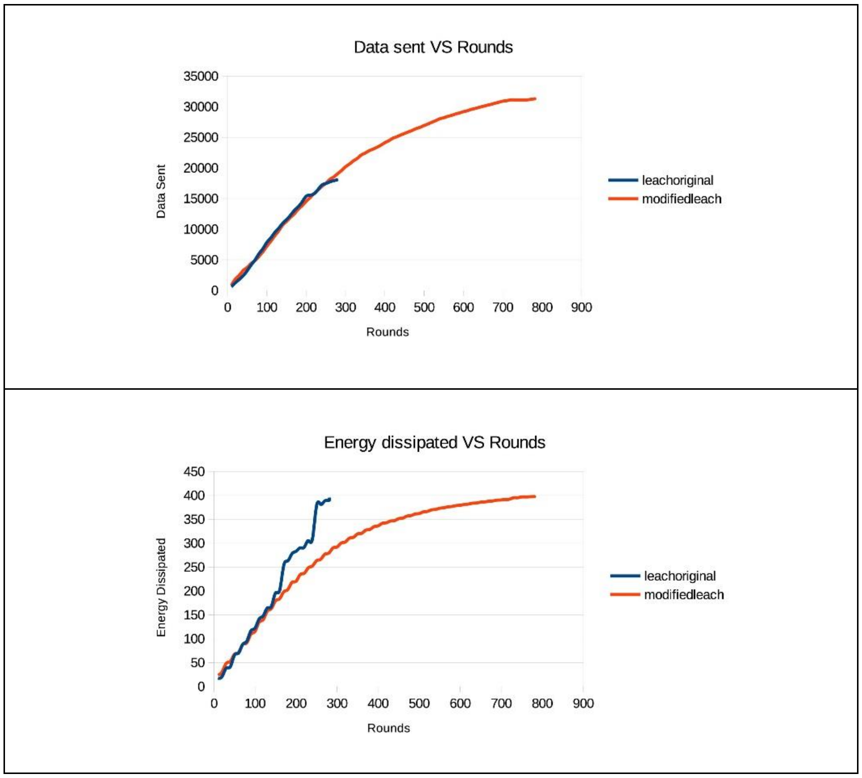


Figure 6.5: Plot showing alive, data transmitted, and energy spent versus rounds (200 nodes with 2J each node). [Original LEACH to LEACH modified]

No. of alive nodes in LEACH-C variable clustering	**No. of clusters in LEACH-C variable clustering**	**No. of alive nodes in LEACH-C**	**No. of clusters in LEACH-C**
100	5	100	5
91	5	82	5
81	5	73	5
73	5	70	5
70	6	70	5
60	6	60	5
51	6	52	5

Table 6.6: A comparison showing the variation in the number of cluster heads in variable LEACH-C and LEACH-C

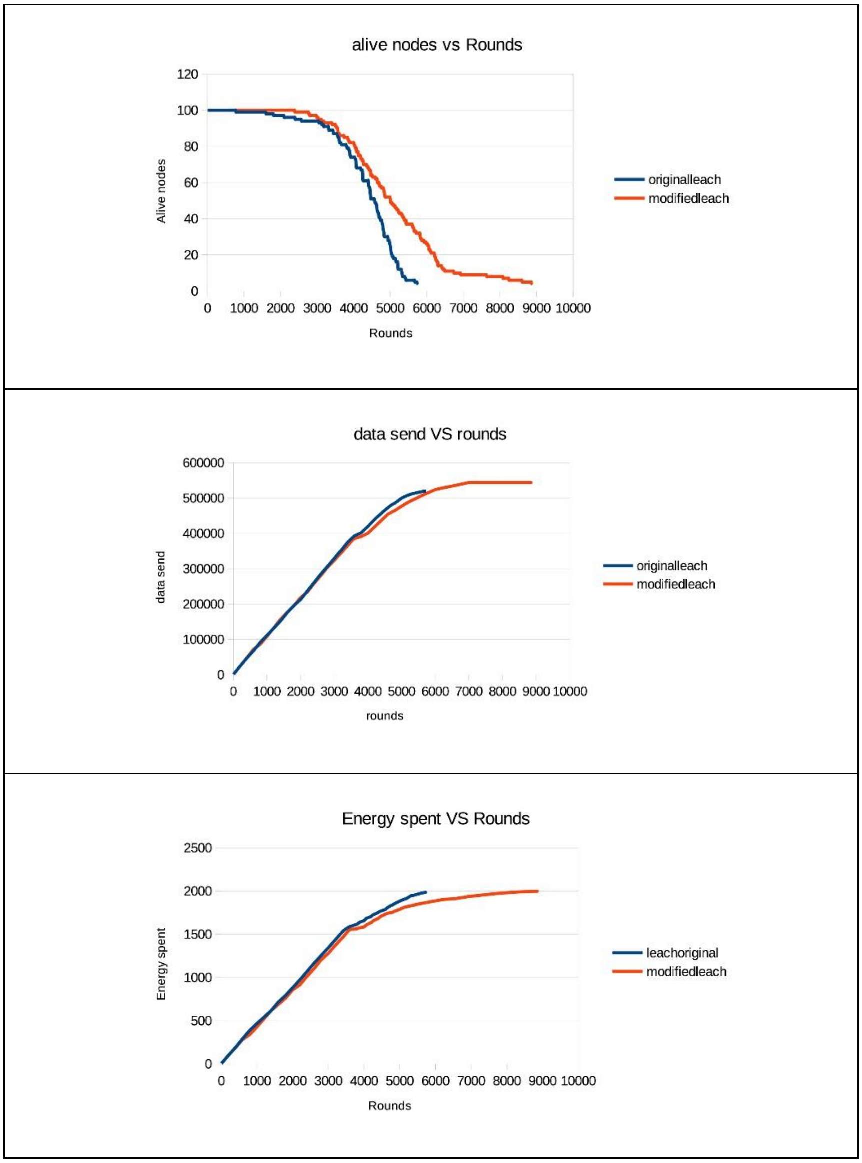


Figure 6.6: Plot showing alive, data transmitted, and energy spent versus rounds (100 nodes with 20J each node). [Original LEACH to LEACH modified]

6.5. Discussions

The above experiments suggest that the network evolution model provides another way of optimizing the performance of the system. It can be easily concluded from section 2.2 that the optimum number of clusters can also be obtained from this evolution model without the use of communication energy principles. Another important aspect of the Network Evolution Model is the optimizing matrix which seeks global optimization rather than local. This matrix, which forms the basis of energy efficient cluster variation, lays the possibility of other variations. The main contribution of this framework is the management of clusters after the death of fifty percent of the nodes. The effectiveness of this framework is strongly confirmed in the case of a large number of nodes compared to a small number. Therefore, this framework can serve as a useful tool in driving the efficiency of the protocol without too much change to the algorithm.

Figure 6.7 shows the relationship between the number of (cluster-alive) nodes spaced through LEACH-C, variable clustering LEACH-C and modified LEACH-C.

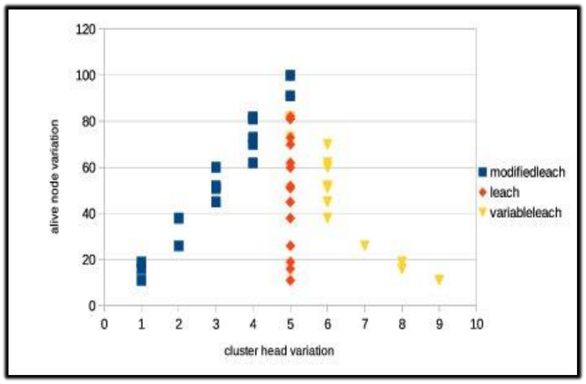


Figure 6.7: The Cluster Head Vs Alive Nodes

100 nodes 20 J/node	**LEACH Modified**	**LEACH-C Modified**
Network lifetime	62%	23%
Data Transferred	4.6	2%
Energy dissipated	Same	same

200 nodes, 2 J/node	**LEACH Modified**	**LEACH-C Modified**
Network lifetime	178%	62%
Data Transferred	72%	23%
Energy dissipated	Same	same

Table 6.7: Percentage of various performance parameters for modified LEACH and LEACH-C.

From Table 6.7 we can easily see that the framework has a larger effect on the performance of the LEACH protocol compared to the LEACH-C protocol due to the formation of less stable clusters in the original LEACH. Therefore, the use of an optimization matrix provides an easy transition from one clustering setup to another as seen in Figure 6.8 (the green line). Moreover, it can be observed that as the number of nodes increases, the performance in terms of network lifetime and data transfer improves. The efficiency boost of 72% and 178% of data transferred and network lifetime is one of the best performances among the protocols developed around LEACH.

6.6. Conclusions and Future Work

The experimental simulation described in this chapter indicates that the fixed type of clustering which exists in LEACH, LEACH-C and similar protocols forbids performance at a system level. This framework is successful in understanding the clustering dynamics of a system; it predicts the efficient scheme of clustering, which is dependent on taking the total number of nodes into consideration rather than the geometry of the field or the electronics used in sensor nodes.

The variable clustering schema, the network lifetime in the case of LEACH and LEACH-C, the electronics of the sensor node and the geometry of the test field could be used to find out how efficiently the perfect clustering schema could be implemented. Future work could be in the direction of a system with soft computing-based cluster forming protocols.

Chapter 7
Effect of Gossiping on Some Basic Wireless Sensor Network Protocols

It has been found that almost all routing protocols suffer from message complexity. To overcome this complexity, we use various optimization processes. One such process is gossiping, where each node forwards a message with some probability, to reduce the overhead of the routing protocols. In this chapter it is shown how gossiping can be used to improve the performance of protocols such as angular routing and spanning tree protocol. It can be seen that in gossiping frameworks the best performance is obtained when the probability of gossiping is 0.6 to 0.9. It can also be seen that there is a reduction of around 25% in the message generation during the execution of the protocols for 100 node random or grid-based networks.

7.1. Introduction

Distributed systems, such as those of ad hoc networks, do have difficulty in having the nodes reach an agreement. As a result, during the execution process, there is a surge of unnecessary messaging in the network, which in turn increases the traffic of the network and a loss of energy is indicated.

In order to mitigate this issue, a gossiping framework is adopted because, as it provides in-network processing in sensor networks, there is no bottleneck or single point of failure, and they do not require any specialized routing and are resistant to failure in unreliable wireless network conditions [127, 80, 110].

Gossip's inherent robustness comes from this random communication pattern that routes new messages around both communication and process failures. Much like real-life gossip- and epidemic-spreading, information disseminated by a gossip protocol spreads quickly and reliably with high probability [112].

Gossip-based algorithms to achieve consensus over a set of agents were initially introduced by Tsitsiklis [7] and have recently received renewed attention from several other researchers [25, 156, 143]. Gossip algorithms for in-network processing have primarily been studied as solutions to consensus problems [32], but until recently they have been used for optimization purposes.

To understand the gossip framework, let us assume the gossip probability is p, and let Pr be the fraction of the nodes that receive messages and Ps be the fraction of nodes which successfully run the gossip executions. Therefore, the fraction that fail to run gossip executions is 1-Ps; similarly, the fraction that do not receive the gossip messages is 1-Pr. It is interesting to note that the case of Pr =1 is only considered. Ideally the fraction of executions Ps where the gossip dies out should be relatively low, while also keeping the gossip probability p low is considered to reduce the message overhead for the implementation of the gossip framework.

In [153] the authors apply gossiping to ad hoc unicast routing. In their work, the authors try to ensure that messages are delivered with conformity even if there is no connected path between the source and the destination at any given point in time. As long as there exists a path due to communication links at some point in time, messages can be delivered through random pair-wise exchanges among mobile hosts.

In [125] the authors use a gossiping framework to improve multicast reachability in ad hoc networks; the authors do not use gossiping to reduce the number of messages sent. In contrast the authors start with an arbitrary, possibly unreliable, multicast protocol to multicast a message. They then use the gossiping framework to randomly exchange messages between nodes in order to recover lost messages. Their premise of assumption is that the routes are known.

In [142] the authors put forward ideas that can reduce the redundancy of the messages used in broadcasting. One of the ideas described is that of a gossiping framework. However, they do not study the properties of a gossiping framework and their implications.

This chapter is divided into six sections. The section 7.1 is a brief introduction to the gossip framework; the section 7.2 deals with a brief review of the protocols on which the gossip framework will be applied; the section 7.3 gives a list of assumptions and the theoretical framework of gossiping; the section 7.4 deals with simulator specifics built on MATLAB; the section 7.5 deals with results and discussions regarding the data generated by the simulator; and the section 7.6 offers conclusions to the chapter and suggestions for future work.

7.2. Brief Overview of Routing Protocols

Angular Routing — The first category of routing protocols are the multi-hop flat routing protocols. In static flat networks, each node definitely plays the same role and sensor nodes help in the sensing activities. Due to the large number of such nodes, it is not feasible to assign an identifier with respect to position and energy. This consideration has led to data-centric routing, where the sink or the base station sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is being requested through queries, attribute-based naming is necessary to specify the properties of data. In addition to the flooding mechanism another scheme adopted for experimental purposes is the introduction of the angle to which the flooding will occur. The relative merits and demerits with respect to flooding are yet to be determined.

Spanning Tree Protocol: Spanning Tree Protocols are the most common routing trees in the networks and particularly in WSNs, because they represent the routing structure in terms of time and energy. These time and energy representations are an efficient means of description. There are many algorithms for the construction of Spanning Tree protocols, including those of Bellman–Ford and Dijkstra. Moreover, WSNs in general do not have centralized management and fixed infrastructure. The authors Bertsekas and Gallager proposed a distributed asynchronous version of the Bellman–Ford algorithm for distributed systems such as wireless sensor networks. This variant of Bellman–Ford (BF) presents fast convergence, in the absence of any synchronization overhead and easy adaptation to topological network changes, but it is not efficient for networks composed of a large number of nodes or dense networks, because of the excessive number of messages required for the tree construction. This is the area where we intend to use gossiping frameworks to reduce the number of messages.

7.3. Assumptions and Theoretical Background

Any protocol that guarantees certain properties has to make certain valid assumptions. However, if the assumptions are explicit then it becomes the responsibility of the developer to satisfy the assumptions. These assumptions are mostly network latency and bandwidth, processing time, failures, and so on.

Therefore, in the premise of gossiping frameworks the following are the assumptions:

1. Participants gossip with one or more partners at fixed time intervals.
2. There is a bound on how many updates are concurrently propagated.

3. Every gossip interaction is independent of concurrent gossiping between other processes.
4. Any two processes can discover each other independently of the gossip mechanism.
5. Processes select gossip partners within a round in an unpredictable random-like fashion.

Based on these assumptions we proceed with the theoretical definition of a gossip framework [7].

We start by considering the source or the start point as the point of initiation. Let the source send the route request with probability 1. When the node receives the route request then it can accept it with probability p and disregard with probability 1-p. However, if the same node receives the request again, it rejects this request. So, a node can broadcast the route request only once. This is the simple representation of a gossip framework and is denoted by G(p).

But the above representation of a gossip framework has some problems with initial conditions. If there are fewer neighbours, then the gossip should die out. In order to counter this situation we have to include hops in the basic definition. Thus, we rewrite as the first k hops before continuing to gossip with probability p. Therefore the gossip framework is G(k,p).

In this communication we focus on the calculation of G(k,p) on various protocols and their effect on different topology properties.

7.4. Simulator Specifics

In order to test the gossiping framework a probability-based environment to simulate the nondeterministic nature of the communication channel and the low-level communication protocol of the motes has been developed using MATLAB. We can visualize the underlying topology that is presented to the protocols and include an arbitrary number of nodes. In this simulator we have designed a probabilistic radio channel model to encounter the non-deterministic nature of the radio channel. This framework can also accommodate an arbitrary number of nodes, an arbitrary (possibly dynamic) topology, and it was designed so that it can easily be embedded into optimization algorithms.

A simplified but fairly accurate model of the MAC layer is designed for the simulator. In the application layer, event-based methods and functions are written on the line of TinyOS environment. The optimization framework is the core to the development of the gossip framework. The main idea behind these methods is some kind of exploration of the error surface; either a gradient-based method, Monte-Carlo search, or a simulated annealing method [143]. The error function of the optimization framework can have any performance metric defined in the parameter space, such as time, energy, throughput or any combination thereof. Due to the stochastic nature of the environment some a priori knowledge is required on the error surface for the convergence of the algorithm. The better the a priori information, the better the convergence of the error surface. To overcome the noisy error surface the simple brute force algorithm is applied to scan the parameter space on a finite grid and thus the optimum value can be found. The detailing of the radio propagation model, the MAC layer and the optimization frame is discussed in Appendix D. The user may supervise the required number of experiments, so that the error surface can gradually improve. Before using the simulator, some of the terminology employed should be defined, such as:

Probability: This is the chance of a neighbour being selected for the reception of the broadcast messages.

Settling Time: This is the time required by the protocol to execute all its methods and functions on the given set of nodes.

Below we depict some of the screen shots of the simulator used. In Figure 7.1 we can see the sensor nodes in blue rectangular boxes. S is the source and D is the destination. We can see that the arrows emanating from the source are directed towards the destinations by the method of broadcasting.

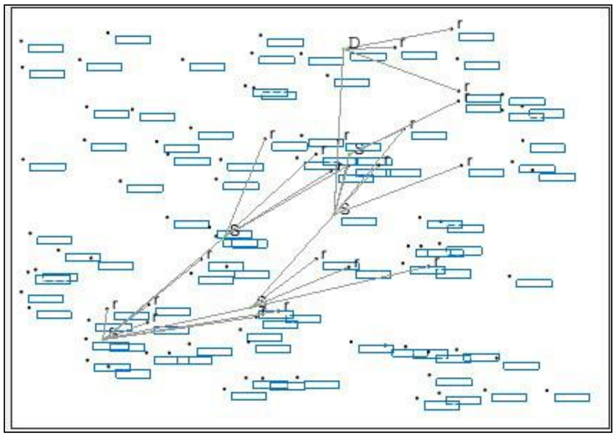


Figure-7.1. The screen shot of angular routing in the simulator display window.

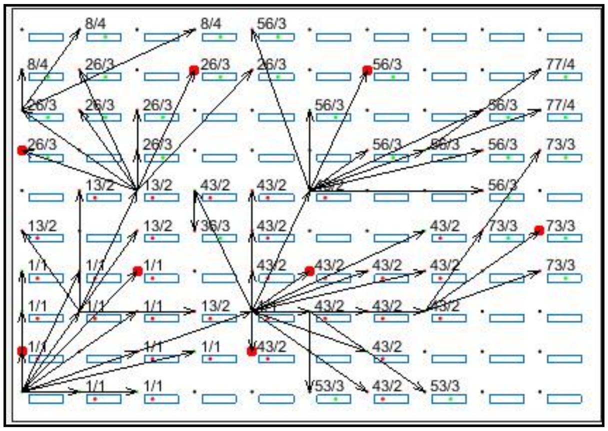


Figure 7.2. The screen shot of the Spanning Tree Protocol in the simulator display window.

7.5. Results and Discussions

The Simulator as described is used with two kinds of protocol

1. Angular Flooding
2. Spanning Tree Protocol

We have used the simulation results of angular flooding with and without a gossiping framework. (Fig-I)

Below we present the table of results.

Messages	Nodes	Settling Time
495	96	0.455
659	96	0.559
379	94	0.382
455	93	0.416
303	83	0.286
384	95	0.4
489	98	0.402
371	84	0.328
304	79	0.376
309	79	0.382

Table 7.1 The angular flooding without gossip framework.

Message	Nodes	Settling Time	Hops
17	22	0.024025	1
62	32	0.0797	3
39	28	0.05445	2
17	16	0.024025	1
24	23	0.024025	1
110	69	0.14465	6
62	39	0.0869	3
73	51	0.0833	3

13	12	0.024025	1
36	26	0.053225	2

Table 7.2 The angular flooding with gossip framework G (0.1, hops)

In Table 7.1 we find the number of nodes involved in angular flooding is definitely greater than what we see in Table 7.2. But the objective of flooding from the source to the destination is not fulfilled by considering G (0.1, hops). As a result, we see that the flooding seems to become controlled in the situation presented in Table 7.2.

Now, as we increase the probability we are certain to find that there is a success in the dissemination of packets from source to destination.

Below we shall depict the Tables for G(0.2, hops) and G(0.3, hops) to justify our case

Message	**Nodes**	**Settling Time**	**Hops**
141	67	0.143	8
105	68	0.11413	5
88	44	0.1158	4
83	37	0.11595	4
22	21	0.024025	1
258	85	0.26015	13
181	76	0.1598	8
287	81	0.29188	14
127	68	0.17465	6
270	76	0.31917	15

Table 7.3 The angular flooding with gossip framework G(0.2, hops)

In the Table 7.3 scheme we have only 50% of the protocol executions meeting the conditions for source to destination.

Message	**Nodes**	**Settling Time**	**Hops**
192	77	0.21243	8
23	22	0.024025	1
98	61	0.11323	5
294	87	0.2989	17
73	47	0.08645	4
229	91	0.1972	14
234	82	0.22585	13
96	43	0.1059	6
230	81	0.2197	13
96	62	0.11613	8

Table 7.4 The angular flooding with gossip framework G(0.3, hops)

Similarly, in the Table 7.4 scheme we have only 60% of the protocol executions meeting the conditions for source to destination.

One interesting fact that can be noted is that as the number of hops increases the settling time increases and is independent of the number of participating nodes.

So, finally, we choose the condition G (0.6, hops) to justify our result.

Messages	**Nodes**	**Settling Time**	**Hops**
409	84	0.39135	29
327	76	0.3858	23
247	82	0.2377	15
160	73	0.20172	16
340	91	0.3234	25
414	95	0.37833	27
220	77	0.2323	17
468	92	0.35823	31
288	92	0.273	24
437	91	0.38288	30

Table 7.5 The angular flooding with gossip framework G(0.6, hops)

Table 7.5 depicts 100% success in the representation of the task accomplishing source to destination. In this process there is an average message saving of 23% (i.e. less message is generated in the gossiping framework) and a 21% saving in terms of settling time. As the total of nodes in the network is 100 the saving seems not greatly significant. However, it will be substantial when the number of nodes in the network is greater than 500; i.e. in the case of a dense network. Now we use the Spanning Tree Protocol (Figure-7.2) on a grid-based network of 100 nodes and obtain the results with or without a gossip framework.

Messages	Nodes	Settling Time
889	100	0.50938

751	99	0.48013
819	100	0.49712
793	100	0.51095
793	100	0.47242
795	100	0.49262
788	99	0.48862
842	100	0.54755
755	100	0.4902
810	100	0.5169

Table 7.6 The Spanning Tree Protocol without a gossip framework

Messages	**Nodes**	**Settling Time**	**Hops**
599	100	0.50095	96
608	100	0.46885	98
614	100	0.49357	98
623	100	0.511	95
610	100	0.4832	96
618	100	0.5001	97
609	100	0.4588	95
597	98	0.4326	91
622	100	0.4567	92

619	99	0.4677	96

Table 7.7 The Spanning Tree Protocol with a gossip framework G(0.6, hops)

Similarly, we can see that there is a message saving up to 24% while using the gossip framework, and the settling time is lessened by 4%.

The primary difference between the two protocols is that one is used for source to destination routing and the other is used for coverage of the network purposes. Hence, we obtain the contrasting figures in both cases.

In addition to the individual performance of probabilities we provide a snapshot of the performance of various probabilities ranging from 0 to 1.

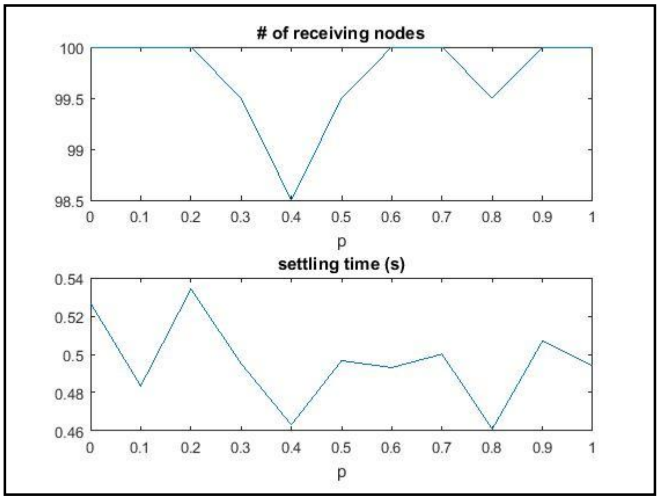


Figure 7.3 The performance of the various Gossip Probabilities for Spanning Tree Protocol.

It can be seen how the receiving nodes settle down to 100 and the settling time is reduced after P=0.6.

Similarly, the same caveat follows from the graph depicted by angular routing protocol.

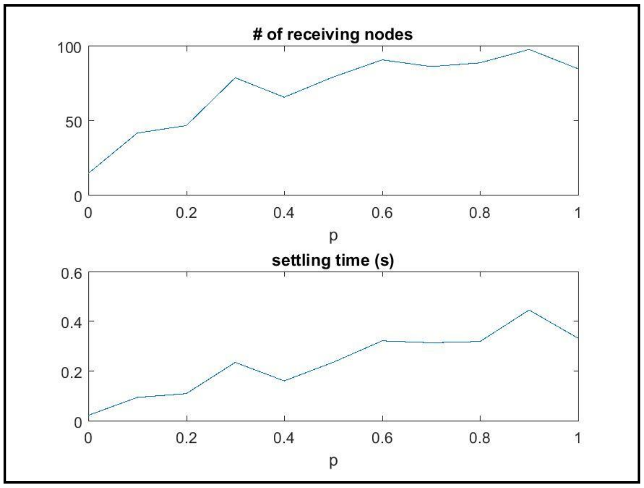


Figure 7.4 The performance of the various Gossip Probabilities for Angular Routing Protocol

7.6. Conclusions and Future Work

As predicted by the theory, the gossiping framework is useful in not only reducing the message generation but also helpful in reducing the throughput time. It can be seen that the savings in message generation is around 25%, but this can increase if the density of the nodes and/or the number of nodes in the network is increased.

In future work, we intend to develop protocols which will have inbuilt gossip frameworks. This is akin to the way the modification of proactive protocols can be done. Because in a proactive routing protocol, every node maintains one or more tables representing the entire topology of the network. These tables are updated regularly using the network information with messages such as the Optimized Link State Routing protocol (OLSR).

Chapter 8
Conclusions and Future Work

8.1 Conclusions

Topology management has been seen as an area of continuous increase in interest during the last decade, the fact that is shown by the amount of published papers and books in this area. The solutions provided in the topology management are mostly theoretical and centralized approaches, which were based on traditional graph theory, which then slowly evolved into the current state with a fully distributed and simple protocols that can run effectively in resource constrained devices. In addition to the offering, connectivity and coverage in the network, is also guaranteed to the area of interest, with different levels of accuracy, which in many cases, without the need of local information of the nodes in the network.

The SOFMTB algorithm reduces the computational overhead by computing the necessary backbone for the protocol (like span tree) under consideration. This makes the routing table manageable in terms of size and computational time. Other advantages include, inherent robustness to node or link failure, changing network geometry (in case of battery depletion), reduced redundant packet transmissions and implicit network reconfiguration. Only disadvantage is the need for sufficient sensor density to maintain network operation. Thus for random distribution of sensor nodes a definite energy efficient topology building algorithm has been developed. SOFMTB it is very easy to implement topology building algorithm which rather becomes a complex one using graph theory. The perseverance of the topology will depend on the number of times the SOFMTB algorithm will be called during the operation of the network.

For hierarchical topology network, production rules are implemented in the context of residual energy and the number of neighbors to find the cluster head of such network. This knowledge based approach was found to perform better than some of the well-known protocols such as A3. In another approach, during initialization process the local information about nodes is used to reduce the maximum power to discover the neighboring nodes. This local information based approach was found to perform better than EECDS (Energy Efficient Connected Dominating Set Algorithm).

For generalized approach towards topology building, the concept of complex networks is used to deduce network evolution model. This framework is formally used to model tree based clustering algorithms in WSN. This formalism theoretically predicted the parameter such as number of cluster heads when the average degree of the node was given for a network. When this prediction was used on the alive nodes in the network implementing LEACH and LEACH-C using optimizing matrix, the network life time efficiency bettered by 178 % and 62 % and that of data transfer rate bettered by 72 % and 23 % respectively. The performance regarding the network life time and the data transfer in case of LEACH is one of the best among the protocols developed over LEACH.

In general the topology building process generates messages, the number of messages depends on functions of the implemented protocol. Thus excess message generated creates computational complexity and leads to drainage of the unnecessary energy. To address this perseverance issue for topology management, gossiping framework is implemented for basic protocols like span tree and angular routing where there is a reduction of around 25% in the message generation during the execution of the protocols for 100 node random or grid-based networks.

Thus with the above mentioned schemes as presented in this dissertation is able to reduce the energy consumption while keeping the important characteristics of network lifetime, connectivity and coverage.

8.2 Future Work

As expected, the following dissertation cannot cover all possible insights from all the directions presented. Some of the areas in which there could be some extension are the following:

1. The most critical drawback of SOFMTB algorithm is that it fails to address synchronization issues which is one of the important aspect for topology survival. This algorithm becomes ineffective when the node density is reduced. This allows researchers to exploit the above mentioned aspects along with other flavors of neural network based on unsupervised to supervised learning.
2. While mentioning the production rules, it is emphasized on the linear combination of the node characteristics viz. residual energy and the number of neighbors. But there can be other node characteristics like average distance of the neighbors from a node, node type and make, power efficiency, scalability, responsiveness, reliability and mobility which can be combined using production rules throws an open problem to the theory of production rules itself for combining many characteristic as a single equation.
3. In the study of network evolution model, the anti-preferential attachment factor has some influence on the network building dynamics. The statements like whether the anti-preferential attachment is complement to preferential attachment and the validity of the routing protocols on extent of variation of anti-preferential attachment factor are very fundamental questions to be researched. On the application of network evolution model via optimization matrix there is considerable success is boosting the network life time and the data transfer over the network for LEACH type protocols, so the challenge remains to implement the optimization matrix for other class of protocols.
4. With the implementation of gossiping framework there has been considerable success in reducing the message complexity by 25% for some well- known basic protocols. As because for a proactive routing protocol, every node maintains one or more tables representing the entire topology of the network. These tables are updated regularly using the information of network using messages like Optimized Link State Routing protocol (OLSR). So there remains an open research problem to implement

gossip framework in proactive protocols to reduce the message complexity which in turn will be useful in managing the energy of the network.