Optimizing MANETs Network Lifetime Using a Proactive Clustering Algorithm

Alyaa Abdulmunem M. Al-Najjar[1], Haitham Shiaibth Chasib[2], Israa Jaber Khalaf AL-OGAILI[3]

[1]College of Administration and Economics, University of Babylon , Iraq
[2]Department of information technology, General Directorate of Education in Babylon, Ministry of Education , Iraq
[3]Department of information technology, General Directorate of Education in Babylon, Ministry of Education , Iraq
[1] alyaaalnajar@gmail.com,[2] haitham.aljabry2014@gmail.com,[3] israa.jaber88@gmail.com

Abstract : In wireless sensor networks that consist of a number of power constrained sensor nodes, the foremost challenges are the limited energy and system lifetime. Therefore, designing efficient routing protocols, which prolong the network lifetime, is one of the most critical issues. This paper evaluated several clustering algorithms, namely: Highest Degree Clustering Algorithm (HDCA), and Lowest Identifier Clustering Algorithm (LIDCA) under three metrics: throughput, Packets Delivered Ratio Factor (PDR) and network lifetime. One of the most important challenges facing Mobile Ad hoc Networks is saving energy that led to a longer network lifetime, which is why we proposed a new clustering algorithm that is considered to be more efficient under network lifetime, and it compared to the clustering algorithms mentioned above. Our experiment occurrences showed that the proposed clustering algorithm supplied a relatively better network lifetime and a more efficient energy distribution for the nodes.

1. Introduction

Self-designing structures of haphazardly moving nodes set up Mobile Ad-hoc Networks (MANETs) in which moving nodes function as mobile terminals, just as directing stations [1]. Ad-hoc networks are partitioned into wireless sensor networks and MANETs. They are called MANETs in light of the autonomy and mobility of their nodes. The significant difficulties that exist in MANETs incorporate the absence of communication infrastructure, the presence of dynamic topology, and the adjustments in the level of association of nodes after some time, which brings about high energy consumption [2].

Besides, clustering is a successful procedure for handling the scalability and dynamics in an enormous scope of mobile ad-hoc networks (MANETs). Also, clustering is frequently utilized for diminishing the network overhead so as to expand the lifetime of the nodes within the network. The nodes in MANETs could then be physically assembled in a similar manner, determined by their logical relationship or interests[3]. Moreover, clustering strategies are utilized in a decent variety of applications, such as ad-hoc networks and data mining. Various clustering techniques have been planned in the prior period, each having its compensation and hindrances [4]. Furthermore, clustering schemes can utilize a lot of clustering metrics, like weighted clustering that utilizes four types of clustering metrics, namely the transmission power, node degree, residual energy of a node, and mobility. Also, clustering reduces the amount of information utilized to store the network state.

Network nodes in a clustering algorithm are separated into clusters. One of the nodes that is answerable for resource allocation, cluster management, packet transport, and routing in the cluster is chosen as the cluster head. Inside these clusters, nodes that have an immediate two-path connection with a single cluster head are called normal nodes or cluster member nodes, whereas nodes that have an immediate two-path connection with more than one cluster head are named gateway nodes, as the latter are utilized in inter-cluster communication [5],[2].

The CH is responsible for gathering data from its members through intra-cluster communication. Also, it could help out different CHs to report data to a BS based on the same form of communicating, in such a way that the CHs often have similar features as the other cluster members [6], [7]. Moreover, the cluster head is a local organizer within its clusters and implements a change of intra-cluster moving and sending of data[8]. Furthermore, cluster heads maintain information about members within its local cluster as well as its connectivity to neighboring clusters. Nodes maintain routes to their respective cluster heads. The cluster heads reduce the broadcast overhead for determining routes to the destination nodes [3].

Both the cluster member and cluster gateway are connected to the cluster head likewise, however, the former implements intra-cluster communication, whereas the latter implements inter-cluster communication as it can communicate with neighboring clusters [9],[10].

1.1 (WCA) :

A weight value is calculated for each node depending on certain metrics, such as the speed, degree, and energy of nodes. Therefore, this algorithm chooses the minimum weighted node as cluster head [11].

1.2 (LIDCA):

This algorithm searches for the oldest and lowest ID from the current cluster. The node has been selected as (CH) for having the lowest ID. Since it is an identifier-based clustering algorithm, it will first assign a unique id to the assigned node or the nodes which are within one cluster [12].

1.3 (HDCA):

As this algorithm utilizes area data for cluster arrangement, it selects the cluster head from the highest degree node in an area. It is a connectivity-based clustering algorithm, and the degree of a node depends on its distance from others, taking the node with the highest degree. The degree of a node is based on the number of nodes associated with or connected to that node. Whenever progressively many local nodes are associated with the cluster node, the highest degree node increments, after which that particular node turns into the cluster head of that cluster [11],[12].

2. Literature review

Throughout the last decade, several studies have been conducted for the selection of cluster heads (CH) in ad-hoc mobile networks using certain the clustering algorithms such as LIDCA [13],[14], WCA [15], and HDCA[16].

The experimental work presented in [2],[17] mainly dealt with (WCA), and has proven that the battery power is consumed by the cluster head more than ordinary nodes due to its additional responsibilities. Similarly, in WCA the weight of each node is determined by its coverage, mobility, transmission power, and battery power, where the node with the lowest weight among its neighbors is chosen as the cluster head.

As for the research on LIDCA conducted in [18], it showed how this algorithm generates a lot of unused clusters, bottlenecks,and causing network traffic. The clustering here is fast and inexpensive because of the random allocation of ID to the nodes. Besides, in this algorithm; the node with the minimum ID among its neighbors will be selected as the cluster head. To avoid weakness and tiredness; the cluster head nodes are changed at certain intervals along with the changes in the ID of the nodes.

The work on HDCA presented in [19],[12] pointed out that this algorithm creates a group mobility pattern, which is a group of nodes transferring in a similar direction and at the same speed. Moreover, it leads to the reduction of the number of clusters, and thereby supplies stability to network. However, it has been noted that it has e relatively very low throughput value.

A position-based clustering algorithm is proposed in[8],[18] for large-hop vehicular ad-hoc networks, which is ased on the geographic position of vehicles and traffic information regarding the cluster formation for this protocol. Additionally, a maximum distance between members and cluster head is used to manage the cluster size as much as possible. Furthermore, the overhead is high for V2I and V2V communications in this protocol.

As for [20], a packet delivery ratio (PDRSC) is proposed between the candidate set calculation model and source node, which leads to the consideration of the network interface. The ERTO is also produced, which optimizes PDRSC, the degree of relay nodes, and the expected energy consumption.

A static clustering algorithm was proposed in [21] to expand the network lifetime. At first, they examined the energy balancing approach and inferred the guideline for the clustering algorithm, after which they proposed an organization system to disperse CHs and MNs in prior known areas, in contrast to the previous arrangements.

In the work of [22], a Mobile Data Gathering-based, Asynchronous Clustering method is proposed, depending on (ACMDGTM). It enhances the lifetime of the network through diminishing the hot spot problem area in the sensor node. The cluster head was selected in view of the remaining energy as well as the sensor's area. Moreover, the sink node considers the moving time from clusters to sink node, and makes use of the data overflow time. Finally, the network lifetime is drawn out and the energy is used productively.

In [23], the CHs is first selected via PSO, followed by the sensor nodes to the CHs which are assigned based on a Weight Sum Approach (WSA) approach. WSA depends on the distance from the CH to the sink, the CH node degree, the residual energy of the CH, and the distance between the sensor node and the CH.

As for [24], the clustering trouble in WSNs was solved by means of the Firefly Algorithm which is a meta-heuristic algorithm. Numerous parameters have been used, such as energy and distance to discover the most suitable set of CHs among the normal sensor node. To decrease the delay in the network, the distance between the nodes and the sink is regarded as a significant parameter.

In the work of[25], the network lifetime is enhancement according to the recommended novel algorithm. This algorithm is based on the solid design recommendations installed in [26]. The central processing entity or the

sink node dynamically improve the communication activity levels of the cellular sensor nodes via the residual energy information mentioned through the sensor nodes, in order to save energy without sacrificing the sensory information data throughput.

3. Research method

A virtual work environment was created to evaluate the performance of each selected clustering algorithm: Highest Degree Clustering Algorithm (HDCA), and Lowest Identifier Clustering Algorithm (LIDCA) , as well as the proposed clustering algorithm, all regarding three metrics, namely throughput, Packets Delivered Ratio Factor (PDR) and network lifetime. The three scenarios will be clarified in detail afterwards. For each scenario, the simulation program for any of the clustering algorithms was implemented one hundred times. Table (1) shows the parameters of environment for each scenario.

Table 1. The parameters of environment

Parameter	Value
Simulator	NetLogo 6.0.4
Nodes number	100
Nodes type	Symmetric
Battery power	100 / joules
Nods speed	2 m/s
Broadcast range	8 / m
Paus Time	6 / s

3.1 Metrics

3.1.1 Network lifetime:

 It is the time from the moment the first data packet was sent until the first node dies [27].

3.1.2 Packet Delivery Ratio: (PDR)

It can be calculated by dividing the number of packets received by the destination over the number of packets initiated by the source [28].

3.1.3 Throughput:

It is the number of successfully received packets in a unit of time [28].

3.1.4 The base station:

It is located at the center of the network which is common to all regions [29],[30].

4. Results and discussion

In this section, three scenarios will be explained so as to obtain the simulation results when implementing each of HDCA, LIDCA, and our proposed algorithm.

4.1 First scenario

 The first scenario is realized by creating a virtual work environment including the parameters mentioned in Table 1, for the proper stimulation of the performance of Highest Degree Clustering Algorithm (HDCA) for three metrics: throughput, PDR, and network lifetime. The node with maximum number of neighbors is selected to be a cluster head. A node's degree means the number of neighbor nodes. Figure 1 shows how the clustering algorithm works.
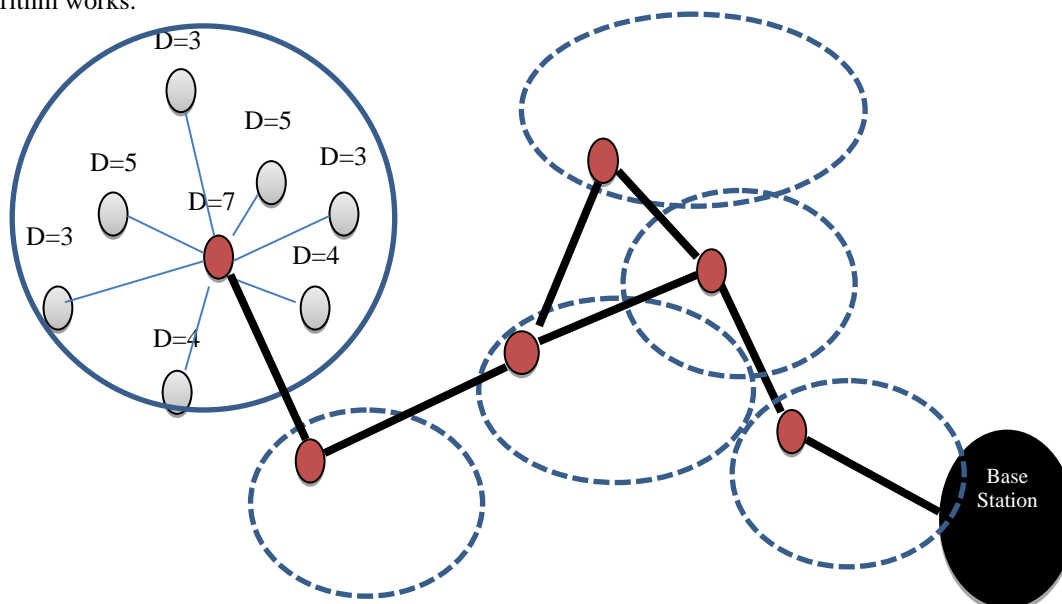
Figure 1. Highest Degree Clustering Algorithm (HDCA)

Table (2) shows the results of implementing the simulation program of HDCA a hundred times.

Table 2. The Simulation Results of (HDCA)

Exp. No.	Throughput	PDR	Network Lifetime	Exp. No.	Throughput	PDR	Network Lifetime
1	0.5	0.1	20	51	1.21052632	0.23	19
2	0.75	0.15	20	52	1.61904762	0.34	21
3	0.6	0.12	20	53	0.39130435	0.09	23
4	1.16	0.29	25	54	1.40625	0.45	32
5	2.0952381	0.44	21	55	1.14285714	0.24	21
6	1.42857143	0.2	14	56	1.33333333	0.28	21
7	0.83333333	0.1	12	57	1.03703704	0.28	27
8	0.93333333	0.28	30	58	0.31818182	0.07	22
9	0.47058824	0.08	17	59	1.52941176	0.52	34
10	1.52631579	0.29	19	60	3.125	0.5	16
11	1.85	0.37	20	61	2.95454545	0.65	22
12	3.28571429	0.69	21	62	0.84210526	0.16	19
13	2.14814815	0.58	27	63	2.09090909	0.23	11
14	1.29032258	0.4	31	64	0.61904762	0.13	21
15	0.7826087	0.18	23	65	0.79166667	0.19	24
16	2.55	0.51	20	66	1.66666667	0.15	9
17	3.26315789	0.62	19	67	1.26315789	0.24	19
18	1.07692308	0.14	13	68	0.68421053	0.13	19
19	2.21052632	0.42	19	69	1.20833333	0.29	24
20	2.125	0.34	16	70	0.25	0.04	16
21	1.16666667	0.21	18	71	0	0	13
22	0.76666667	0.23	30	72	4.47058824	0.76	17
23	1.94444444	0.35	18	73	1.79166667	0.43	24
24	0.8	0.16	20	74	0.25	0.06	24
25	1.9	0.	30	75	1.8965517	0.	29

				75	2	0.55	
26	2.25	0.36	16	76	1	0.22	22
27	0.66666667	0.12	18	77	2.88888889	0.26	9
28	0.89473684	0.17	19	78	1.5	0.3	20
29	0.76190476	0.16	21	79	3.14285714	0.44	14
30	0.6875	0.11	16	80	1.36	0.34	25
31	0.81481481	0.22	27	81	1.65217391	0.38	23
32	1.5625	0.25	16	82	0.56521739	0.13	23
33	1.31818182	0.29	22	83	1.04545455	0.23	22
34	2.46666667	0.74	30	84	1.83333333	0.44	24
35	3.375	0.54	16	85	1.7826087	0.41	23
36	1.95833333	0.47	24	86	1.16666667	0.21	18
37	1.33333333	0.32	24	87	1.96	0.49	25
38	0.85714286	0.12	14	88	1.28571429	0.18	14
39	1.4	0.35	25	89	1.4	0.21	15
40	1.90909091	0.42	22	90	3	0.39	13
41	0.76	0.19	25	91	1.05555556	0.19	18
42	1.9047619	0.4	21	92	1.32258065	0.41	31
43	2.42857143	0.51	21	93	2.73684211	0.52	19
44	3.38461538	0.44	13	94	2.68181818	0.59	22
45	0.5	0.13	26	95	1.42857143	0.3	21
46	1.22222222	0.22	18	96	2.61538462	0.34	13
47	2.38709677	0.74	31	97	0.25806452	0.08	31
48	1	0.13	13	98	1.125	0.27	24
49	2.375	0.57	24	99	1.27272727	0.28	22
50	1.57142857	0.33	21	100	1.78125	0.57	32

4.2 Second scenario

The second scenario is realized by creating a virtual work environment in light of the parameters referred to in Table 1, so as to simulate the performance of Lowest Identifier Clustering Algorithm (LIDCA) in three metrics: throughput, PDR, and network lifetime. The oldest nodes in the network is selected to be the cluster head. Figure 2 shows how the clustering algorithm works.
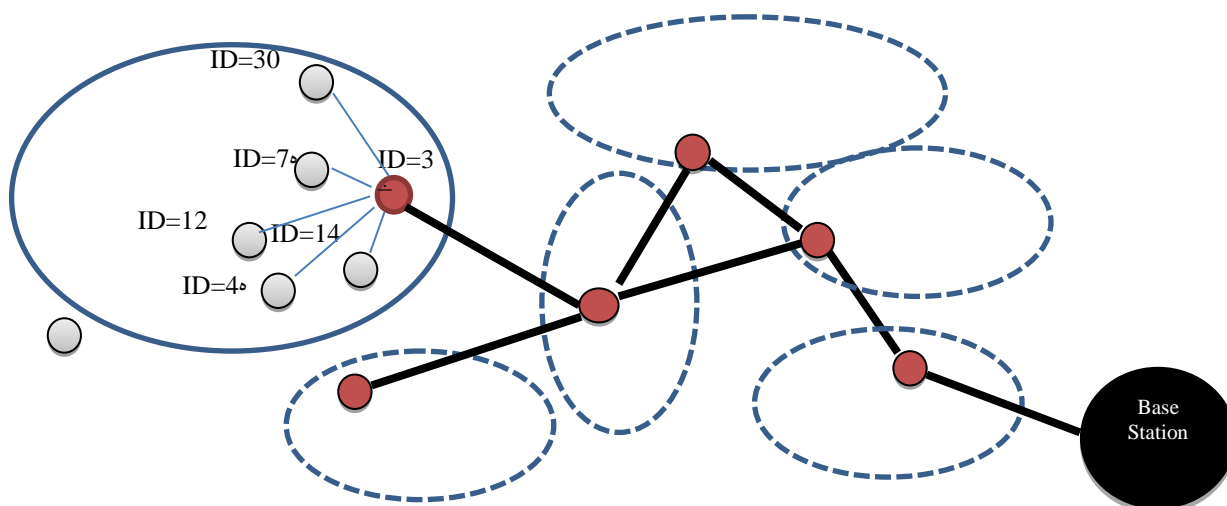
Figure 2. Lowest Identifier Clustering Algorithm (LIDCA)

Table (3) shows the results of implementing the simulation program of LIDCA in one hundred experiments.

Table 3. The Simulation Results of (LIDCA)

Exp. No.	Throughput	PDR	Network Lifetime	Exp. No.	Throughput	PDR	Network Lifetime
1	1.210526316	0.23	19	51	1.736842105	0.33	19
2	0.764705882	0.13	17	52	0.842105263	0.16	19
3	0.842105263	0.16	19	53	1.631578947	0.31	19
4	1.421052632	0.27	18	54	0.631578947	0.12	19
5	0.611111111	0.11	18	55	0.684210526	0.13	19
6	1.157894737	0.22	19	56	1.421052632	0.27	19
7	0.526315789	0.1	19	57	1.421052632	0.27	19
8	0.421052632	0.08	19	58	0.631578947	0.12	19
9	1.105263158	0.21	19	59	1.105263158	0.21	19
10	0.368421053	0.07	18	60	1.117647059	0.19	17
11	0.555555556	0.1	18	61	1.684210526	0.32	19
12	0.526315789	0.1	19	62	0.578947368	0.11	19
13	1.166666667	0.21	18	63	0.315789474	0.06	19
14	0.894736842	0.17	19	64	1.235294118	0.21	17
15	0.578947368	0.11	19	65	2.105263158	0.4	19
16	0.842105263	0.16	19	66	1.736842105	0.33	19
17	0.210526316	0.04	19	67	1.055555556	0.19	18
18	0.789473684	0.15	19	68	0.842105263	0.16	19

19	0.631578947	0.12	18	69	0.631578947	0.12	19
20	0.473684211	0.09	19	70	1.947368421	0.37	19
21	1.117647059	0.19	17	71	0.631578947	0.12	19
22	1.052631579	0.2	19	72	1.555555556	0.28	18
23	1.666666667	0.25	15	73	0.611111111	0.11	18
24	0.315789474	0.06	19	74	0.947368421	0.18	19
25	0.470588235	0.08	17	75	0.526315789	0.1	19
26	0.842105263	0.16	19	76	1.555555556	0.28	18
27	0.789473684	0.15	19	77	1.555555556	0.28	18
28	0.789473684	0.15	19	78	0.684210526	0.13	19
29	1.473684211	0.28	19	79	0.722222222	0.13	18
30	1.052631579	0.2	18	80	0.526315789	0.1	19
31	1.166666667	0.21	18	81	0	0	19
32	0.368421053	0.07	19	82	2.210526316	0.42	19
33	1.529411765	0.26	17	83	0.375	0.06	16
34	1.421052632	0.27	19	84	1.368421053	0.26	18
35	0.105263158	0.02	19	85	1.421052632	0.27	19
36	0.0625	0.01	16	86	0.789473684	0.15	19
37	1.235294118	0.21	17	87	1.833333333	0.33	18
38	0.947368421	0.18	19	88	1	0.19	19
39	0.388888889	0.07	18	89	1	0.19	19
40	1.315789474	0.25	19	90	0.157894737	0.03	19
41	1.105263158	0.21	19	91	1.736842105	0.33	19
42	0.388888889	0.07	18	92	0.736842105	0.14	19
43	1.888888889	0.34	18	93	1.333333333	0.24	18
44	1.555555556	0.28	18	94	0.473684211	0.09	19
45	0.947368421	0.18	19	95	1.578947368	0.3	19
46	0.157894737	0.03	19	96	0.736842105	0.14	19
47	0.526315789	0.1	19	97	1.684210526	0.32	18
48	2.5	0.45	18	98	0.444444444	0.08	18

| 49 | 1.1578947 37 | 0.2 2 | 19 | 99 | 1.058823529 | 0.1 8 | 17 |
| 50 | 0.6315789 47 | 0.1 2 | 19 | 10 0 | 1 | 0.1 9 | 19 |

4.3 Third Scenario

The third scenario is creating a virtual work environment with parameters mentioned in Table 1, so as to simulate the performance of the proposed clustering algorithm under the metrics of throughput, PDR, and network lifetime. The main goal of the proposed clustering algorithm is to save energy and thereby enabling the network to function as long as possible. This algorithm depends on two factors in selecting the cluster head node, namely the node battery energy and the node degree. The standard of the node is represented by a simple equation called weight equation, which has two weights, each equaling 0. 5, the sum of which equals 1. The weighted equation can be explained as shown below:

W = 0.5 × Battery Energy + 0.5 × Node's Degrees

The node with maximum weight is selected to be a cluster head. Figure (3) explains the proposed clustering algorithm work.
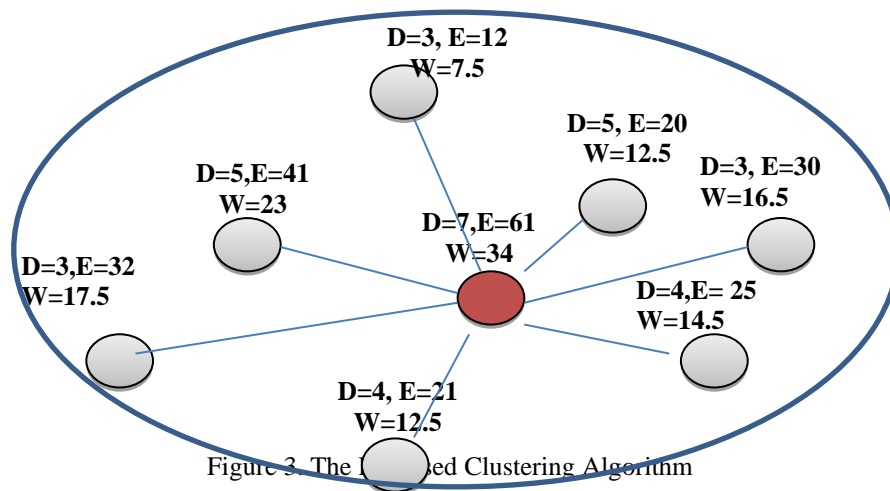

Figure 3. The Proposed Clustering Algorithm

Table (4) shows the results of implementing the simulation program of the proposed cluster algorithm a hundred times.

Table 4. The Simulation Results of The Proposed Clustering Algorithm (100 experiments)

Ex p. No.	Throughpu t	P DR	Networ k Lifetime	Exp . No.	Throughpu t	P DR	Networ k Lifetime
1	0.5238095	0. 11	21	51	1.025641	0. 4	39
2	2	0. 44	22	52	0.56	0. 14	25
3	1.3142857	0. 46	35	53	0.625	0. 25	39
4	0.3684211	0. 14	37	54	0.5	0. 17	34
5	0.9583333	0. 23	24	55	1.08	0. 27	25
6	1.195122	0. 49	40	56	1.4871795	0. 58	39
7	1.425	0. 57	39	57	0.425	0. 17	39
8	0.4473684	0. 17	38	58	1.6176471	0. 55	34
9	1.4642857	0. 41	28	59	1	0. 38	38

10	1	0.41	40	60	1.1481481	0.31	27
11	1.1052632	0.42	37	61	0.6756757	0.25	36
12	1.4210526	0.27	19	62	0.3589744	0.14	39
13	0.6842105	0.26	37	63	1.1621622	0.43	37
14	3.03125	0.97	31	64	0.7727273	0.17	22
15	0.2777778	0.1	36	65	0.4473684	0.17	38
16	1.3	0.52	39	66	0.3888889	0.14	36
17	1.3684211	0.26	19	67	1.6	0.64	40
18	0.2424242	0.08	33	68	1.12	0.28	25
19	0.3888889	0.14	36	69	2.2051282	0.86	38
20	1.025641	0.4	38	70	0.1081081	0.04	37
21	0.5909091	0.13	22	71	0.2941176	0.1	34
22	0.2058824	0.07	34	72	0.5	0.13	26
23	0.6216216	0.23	36	73	0.475	0.19	40
24	0.5277778	0.19	36	74	0.7297297	0.27	37
25	0.75	0.3	40	75	0.75	0.3	39
26	1.5675676	0.58	37	76	1.2432432	0.46	37
27	1.5625	0.25	16	77	2.4117647	0.82	33
28	1.1578947	0.44	38	78	0.9411765	0.32	34
29	0.5526316	0.21	38	79	1.5405405	0.57	36
30	2.0434783	0.47	23	80	0.3846154	0.15	39
31	0.3888889	0.14	36	81	1.2941176	0.22	17
32	2.4166667	0.87	35	82	2.2424242	0.74	33
33	0.3684211	0.14	38	83	0.6486486	0.24	37
34	0.65625	0.21	32	84	0.9210526	0.35	38
35	0.5526316	0.21	38	85	2.9411765	1	33
36	1.3846154	0.54	38	86	0.9736842	0.37	38
37	0.3	0.12	39	87	0.5	0.19	38
38	0.6111111	0.22	36	88	0.4473684	0.17	38
39	0.575	0.23	39	89	1.2608696	0.29	23

40	0.5	0.19	38	90	1.575	0.63	40
41	0.7073171	0.29	41	91	0.9268293	0.38	41
42	1.3611111	0.49	36	92	0.9230769	0.36	39
43	0.5675676	0.21	37	93	0.3125	0.1	32
44	1.5151515	0.5	33	94	0.5483871	0.17	31
45	0.5833333	0.21	36	95	2.025641	0.79	38
46	0.0810811	0.03	37	96	1.4814815	0.4	27
47	1.3703704	0.37	27	97	0.8918919	0.33	37
48	0.4848485	0.16	33	98	1.1842105	0.45	38
49	0.75	0.21	28	99	0.5862069	0.17	29
50	1.4166667	0.51	36	100	0.5925926	0.16	27

5. Algorithms Performance Comparison

A comparison has been drawn among the proposed clustering algorithm and each of the clustering algorithms HDCA, and LIDCA in light of three metrics: throughput, PDR, and network lifetime. Table (5) shows the average values of implementation results for each clustering algorithm.

Table 5. The result comparison for each clustering algotithm.

Clustering Algorithm Name	Network TimeLife Average	PDR Average	Throughput Average
The Proposed Algorithm	33.98	0.3273	0.975678
HDCA	20.96	0.3131	1.530002
LIDCA	18.5	0.181	0.976804

The results listed in Table (5) are represented in Figures (4-6) to illustrate the differences in algorithm performance. These figures show the average values for the throughput, PDR, and network lifetime of the clustering algorithms, respectively.
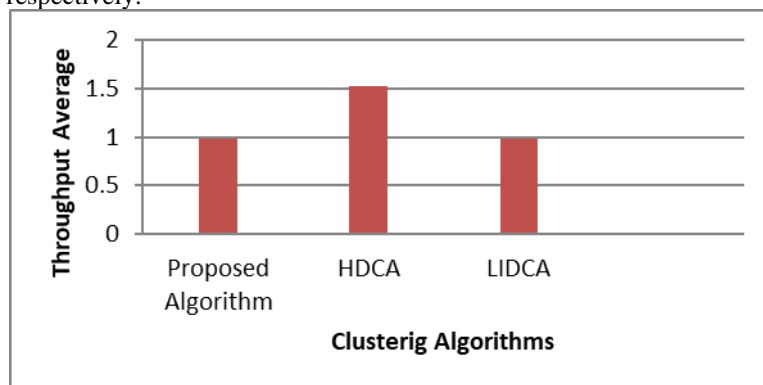

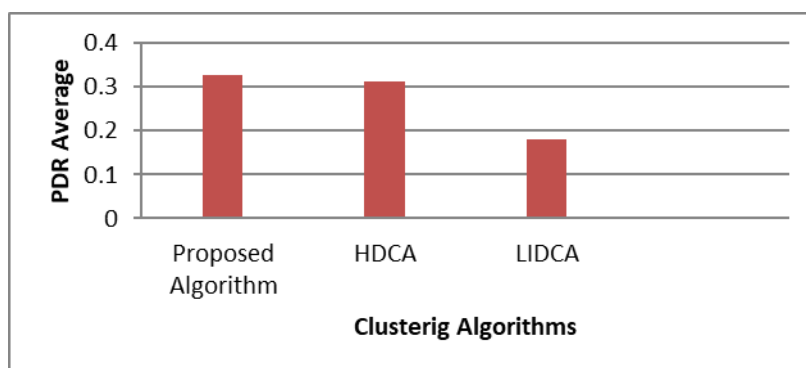Figure 4. The average of throughput
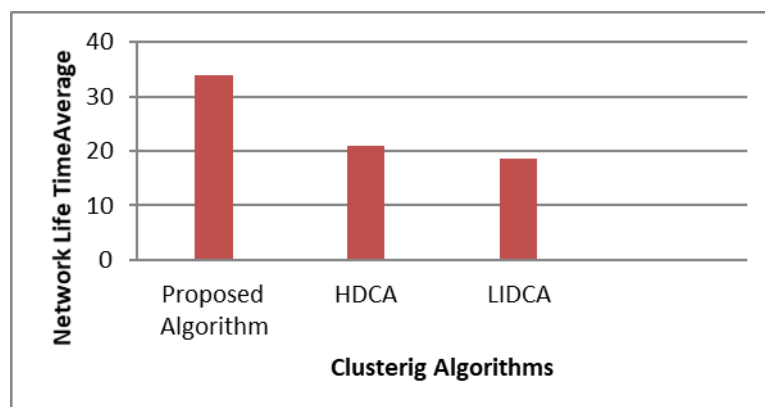
Figure 5. The average of PDR

Figure 6. The average of Network Lifetime

6.   Conclusion

Clustering algorithms make the mobile ad-hoc network work efficiently, as specific nodes called a cluster head node are assigned to be responsible for transmission operations like gathering data and sending them to the goal node (Base Station Node). Therefore, each clustering algorithm has different rules for the selection of a cluster head node, which in its turn leads to different performances, as has been pointed out in this paper. Moreover, the most important challenge for Mobile Ad-hoc Networks is the saving of energy, which prolongs the network lifetime as much as possible, thereby leading to an increase in PDR value. Higher PDR values are associated with lower rates of packets loss.

When evaluating the performance of the selected algorithms, it has been noticed that the performance for High Degree Clustering Algorithm (HDCA) were better than that of the Lowest Identifier Clustering Algorithm (LIDCA) in light of three metrics: throughput, PDR, network lifetime. Therefore, the proposed algorithm included particular criteria for both algorithms in selecting the cluster head. Overall, the merging of criteria for both algorithms is represented by a simple weights equation, where the higher weight node is the most appropriate one to be a cluster head. Therefore, the conclusion can be drawn that the proposed algorithm has offered a lower performance under the metric of throughput, as this algorithm appeared to be rather time-consuming in forming clusters, as compared to the time required for the selected algorithms.

References

1.   S. Z. Hussain and N. Ahmad, "Minimizing broadcast expenses in clustered ad-hoc networks," *J. King Saud Univ. Inf. Sci.*, vol. 30, no. 1, pp. 67–79, 2018.
2.   S. A. Sharifi and S. M. Babamir, "The clustering algorithm for efficient energy management in mobile ad-hoc networks," *Comput. Networks*, vol. 166, p. 106983, 2020.
3.   A. Banerjee, C.-T. King, and H.-C. Hsiao, "On state maintenance in cluster-based mobile ad-hoc networks," *Ad Hoc Networks*, vol. 66, pp. 95–109, 2017.
4.   M. Ramalingam and R. Thangarajan, "Mutated k-means algorithm for dynamic clustering to perform effective and intelligent broadcasting in medical surveillance using selective reliable broadcast protocol in VANET," *Comput. Commun.*, vol. 150, no. July 2019, pp. 563–568, 2020.
5.   M. S. Talib, A. Hassan, Z. A. Abas, M. F. Ali, M. N. Al-Mhiqani, and A. A. Mohammed,

"Clustering in VANETs perspective: Concepts, topology and applications," *Int. J. Adv. Sci. Technol.*, vol. 28, no. 8, pp. 471–484, 2019.

6.   J. E. Z. Gbadouissa, A. A. A. Ari, C. Titouna, A. M. Gueroui, and O. Thiare, "HGC: HyperGraph based Clustering scheme for power aware wireless sensor networks," *Futur. Gener. Comput. Syst.*, vol. 105, pp. 175–183, 2020.

7.   M. S. Talib *et al.*, "A Center-based Stable Evolving Clustering Algorithm with Grid Partitioning and Extended Mobility Features for VANETs," *IEEE Access*, vol. 8, pp. 169908–169921, 2020.

8.   H. Fatemidokht and M. K. Rafsanjani, "QMM-VANET: An Efficient Clustering Algorithm Based on QoS and Monitoring of Malicious Vehicles in Vehicular Ad Hoc Networks," *J. Syst. Softw.*, p. 110561, 2020.

9.   M. F. Khan, K.-L. A. Yau, R. M. D. Noor, and M. A. Imran, "Survey and taxonomy of clustering algorithms in 5G," *J. Netw. Comput. Appl.*, p. 102539, 2020.

10.   F. Abbas and P. Fan, "Clustering-based reliable low-latency routing scheme using ACO method for vehicular networks," *Veh. Commun.*, vol. 12, pp. 66–74, 2018.

11.   P. K. Piyalikar and M. D. Barma, "Forecast weighted clustering in MANET," *Procedia Comput. Sci.*, vol. 89, pp. 253–260, 2016.

12.   M. Gavhale and P. D. Saraf, "Survey on algorithms for efficient cluster formation and cluster head selection in MANET," *Procedia Comput. Sci.*, vol. 78, no. C, pp. 477–482, 2016.

13.   C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE J. Sel. AREAS Commun.*, vol. 15, no. 7, pp. 1265–1275, 1997.

14.   D. J. Baker and A. Ephremides, "A Distributed Algorithm for Organizing Mobile Radio Telecommunication Networks.," in *ICDCS*, 1981, pp. 476–483.

15.   M. Chatterjee, S. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Cluster Comput.*, vol. 5, pp. 193–204, 2002.

16.   M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wirel. networks*, vol. 1, no. 3, pp. 255–265, 1995.

17.   M. Chatterjee, S. K. Das, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," in *Globecom'00-IEEE. Global Telecommunications Conference. Conference Record (Cat. No. 00CH37137)*, 2000, vol. 3, pp. 1697–1701.

18.   D. Gavalas, G. Pantziou, C. Konstantopoulos, and B. Mamalis, "Lowest-ID with adaptive ID reassignment: a novel mobile ad-hoc networks clustering algorithm," in *2006 1st International Symposium on Wireless Pervasive Computing*, 2006, pp. 5-pp.

19.   M. Kumrawat and M. Dhawan, "Survey on clustering algorithms of wireless sensor network," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 3, p. 2046, 2015.

20.   P. Yan, S. Choudhury, F. Al-Turjman, and I. Al-Oqily, "An energy-efficient topology control algorithm for optimizing the lifetime of wireless ad-hoc IoT networks in 5G and B5G," *Comput. Commun.*, 2020.

21.   A. Ghosal and S. Halder, "Lifetime optimizing clustering structure using Archimedes' spiral-based deployment in WSNs," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1039–1048, 2015.

22.   J. Wang, Y. Gao, W. Liu, W. Wu, and S.-J. Lim, "An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks," *Comput. Mater. Contin*, vol. 58, pp. 711–725, 2019.

23.   P. C. S. Rao, P. K. Jana, and H. Banka, "A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks," *Wirel. networks*, vol. 23, no. 7, pp. 2005–2020, 2017.

24.   A. Sarkar and T. S. Murugan, "Cluster head selection for energy efficient and delay-less routing in wireless sensor network," *Wirel. Networks*, vol. 25, no. 1, pp. 303–320, 2019.

25.   D. A. Guimaraes, L. J. Sakai, A. M. Alberti, and R. A. A. De Souza, "Increasing the lifetime of mobile WSNs via dynamic optimization of sensor node communication activity," *Sensors*, vol. 16, no. 9, p. 1536, 2016.

26.   Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Commun. Lett.*, vol. 9, no. 11, pp. 976–978, 2005.

27.   P. Le Nguyen, T. H. Nguyen, and K. Nguyen, "A Path-Length Efficient, Low-Overhead, Load-Balanced Routing Protocol for Maximum Network Lifetime in Wireless Sensor Networks with Holes," *Sensors*, vol. 20, no. 9, p. 2506, 2020.

28.   S. Singh, I. Sharma, P. Saurabh, and R. Prasad, "Fuzzy Logic Based Packet Dropping Detection Approach for Mobile Ad-Hoc Wireless Network," in *Soft Computing for Problem Solving*, Springer, 2020, pp. 263–273.

29.   [29] Clustering for Energy Efficient Wireless Sensor Network," *Wirel. Pers. Commun.*, pp. 1–18, 2020.

30.   A. A. M. Al-Najjar and H. S. Chasib, "Design and implementation weights equation for optimization DSR protocol in MANETs environment," Int. J. Adv. Sci. Technol., vol. 28, no. 8, pp. 457–470, 2019.