Routing in Wireless Sensor Networks: Improved Energy Efficiency and Coverage using Unmanned Vehicles

Long Kim Le *lle3@islander.tamucc.edu*
Computer Science Department
Texas A&M University – Corpus Christi
Corpus Christi, 78412, U.S.A

Ahmed M.Mahdy *ahmed.mahdy@tamucc.edu*
Computer Science Department
Texas A&M University – Corpus Christi
Corpus Christi, 78412, U.S.A

Abstract

This paper proposes a new method for collecting distributed data in Wireless Sensor Networks (WSNs) that can improve the energy efficiency and network coverage; especially in remote areas. In multi-hop communication, sink nodes are responsible for collecting and forwarding data to base stations. The nodes that are located near a sink node usually deplete their battery faster than other nodes because they are responsible for aggregating the data from other sensor nodes. Several studies have proved the advantages of using mobile sink nodes to reduce energy consumption. Nonetheless, the need for compatible and efficient routing algorithms cannot be understated. Accordingly, a hybrid routing algorithm based on the Dijkstra's and Rendezvous algorithms is proposed. To improve the energy efficiency and coverage, Energy Efficient Hybrid Unmanned Vehicle Based Routing Algorithm (E^2HUV) is proposed to create a routing path for Unmanned Aerial Vehicles (UAVs) that can be used as mobile sinks in WSNs. Performance results show that the E^2HUV algorithm offers better efficiency as compared to currently existing algorithms.

Keywords: Unmanned Systems, Wireless Sensor Networks, Mobile Sinks, Scheduling, Routing, Coverage.

1. INTRODUCTION

A WSN can be defined as a network of sensor nodes that are small devices with limited battery capacity which may be difficult and impractical to replace. In addition, they are typically limited in computation, communication and memory capabilities. These characteristics present challenging constraints in designing energy efficient protocols. Much research has focused on the energy efficiency of WSNs emphasizing the energy hole issue [1]. The closer a sensor node is to a sink node, the faster its battery runs out [1]. As a result, the sink node depletes its energy and becomes disconnected from the network. This adversely affects the coverage of the sensor field. Hence, controlling the energy consumption of sensor nodes is critical for the performance and efficiency of WSNs.

Unmanned Aerial Vehicles (UAVs) have been witnessing a rapid growth in applications and capabilities in the last few years. These unmanned systems come in various types, sizes and features. This opens the door for a wide range of applications. A relevant example is the use of UAVs as sink nodes to aggregate data in WSNs in an energy efficient way [2].

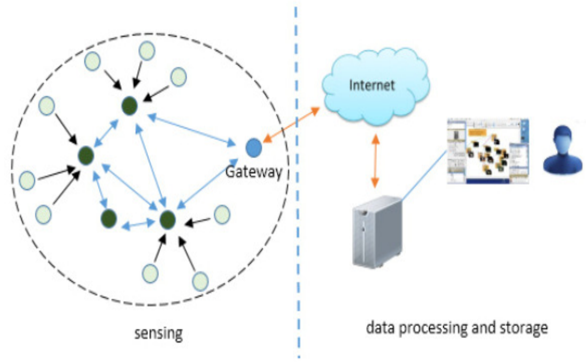

FIGURE 1: WSN infrastructure [9].

2. LITERATURE REVIEW
A. UAVs for WSN
UAVs have recently been used for data collection purposes in WSN. In such contexts, the data is typically forwarded to the base station for analysis purposes. UAVs have been shown to improve the WSN energy consumption issue. They act as mobile sink nodes that can directly impact the overall network energy consumption. The network lifetime and throughput can be increased by implementing an effective data gathering routing scheme in air-to-ground communication network [2]. In addition, there are also factors that can affect the network lifetime such as the communication (cooperative or non-cooperative) between sensor nodes [3].

In adhoc networks, especially WSNs, much research has been done on energy efficient protocols, algorithms, and frameworks. The objectives of such research are to effectively relay the data among sensor nodes, to minimize the energy cost, and to maximize the network lifetime.

Existing protocols using a mobile sink in WSNs can be classified into two categories: 1) direct, where a mobile sink visits each sensor node in an ultimate path and collects data via a single-hop; and 2) rendezvous, where a mobile sink only visits nodes defined as Rendezvous Points (RPs). The main goal of protocols in the first category is to minimize data collection delays by using single hop communication, whereas those in the second category aim to find a subset of RPs that minimize energy consumption but the heavy multi-hop communication issue still exists. In the following, the advantages and disadvantages of these protocols will be discussed.

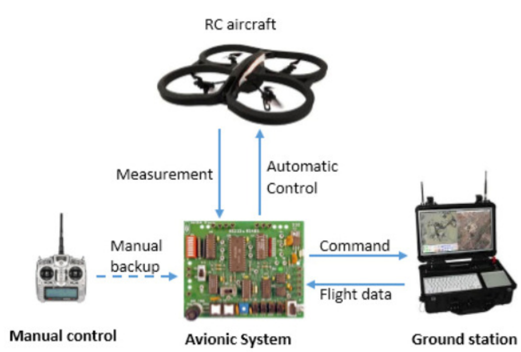

FIGURE 2: Example of a UAS [9].

B. Direct:
Quality of service (QoS) is a major requirement in many applications. In this context, much focus has been dedicated to improve energy consumption. Energy-inefficient algorithms reduce the network lifetime and impact its performance. The energy hole problem is still not solved because

the sensors forward the data to a single sink node that reduces its battery faster [2]. Mobile sink nodes play an important role in efficiently forwarding data to base stations [4].

The three benchmarks of direct routing schemes in WSNs are Dijkstra's algorithm, Low Energy Adaptive Clustering Hierarchy algorithm (LEACH), and Floyd-Warshall's algorithm. Firstly, Dijkstra's algorithm plays an important role in practical applications of data monitoring; it considers the energy and time constraint while sending the message from source node to destination node. In WSNs, this algorithm is used to find the shortest path between pairs of source and destination nodes for data aggregation through all reachable nodes in a sensing field. Dijkstra's algorithm can reduce the network energy consumption significantly [5]. However, this algorithm can be inefficient because of the cost of computations. By applying this algorithm to the mobile sink scenario, the mobile sink will visit each and every node to collect data with single-hop communication. Thus, it can reduce the network energy significantly. In networks with a large number of sensor nodes, this algorithm seems inefficient because of the high data delivery delay.

Secondly, the LEACH algorithm randomly selects some nodes as cluster heads (CHs) for forwarding data and balancing the energy of the network by rotating this role. In this algorithm, sensor nodes will find and only communicate with their closest neighbors (closest nodes) and forward data to base station. By using this algorithm, the energy of the nodes is distributed and the chances of depleting the energy of nodes is less and the power required to transfer data per round is balanced uniformly. As a result, the network lifetime can be increased [6].

Thirdly, Floyd-Warshall's algorithm is a combination of Floyd's algorithm and Warshall's algorithm in order to find the transitive closure of a graph. Floyd's algorithm finds the shortest path but it is not suitable for applying to WSNs. This algorithm finds all the minimum distance between pairs of nodes.

C. Rendezvous:

Instead of letting a mobile sink travel along all the nodes in the network, the use of Rendezvous Nodes (RNs) can reduce the network energy consumption significantly [7]. RNs not only can perform the cluster head role, but also can reduce the network energy cost. The RNs selection algorithm [5] shows that by only visiting the set of RNs, the mobile sink can minimize the travel tour length and avoid the density of the heavy multi-hop communication. As the result, the energy consumption decreases and accordingly the network lifetime increases.

Mobile-sink path selection using weighted rendezvous planning algorithm (WRP) is one of the most common algorithms based on the concept of RNs. This algorithm is utilizing a hybrid moving pattern where the mobile sink node will only visit RNs. Reflecting a set of factors, weights can be assigned to sensor nodes. Based on these weight values, the algorithm chooses a path for the mobile sink. WRP can increase the network lifetime by 44% on average [1]. It also decreases energy consumption by 22% on average [1]. Despite these improvement, the amount of data handled by RNs, especially in large networks, can be overwhelming leading to buffer overflow or high delivery delays when communicating with the mobile sink. Much research has been dedicated to RN-based algorithms aiming to improve the overall performance of the algorithm. One addition is to find the closest RNs among all nodes to create a more efficient travelling path for mobile sinks [8]. By applying this protocol, the mobile sink tour length is significantly decreased compared to traditional RN algorithms.

3. PROPOSED ALGORITHM

3.1. WSN Challenges

WSNs have been studied and researched for many years. Although, a lot of improvements have been proposed and applied over the years, there are still some outstanding challenges that may limit the use of WSNs for certain applications. The two major challenges impacting the WSNs in terms of lifetime, coverage and performance are: power consumption and redundant node requirements. These issues have been a research focus in recent years, yet, the need for better solutions cannot be understated.

In real-life scenarios and for specific applications, a WSN can be deployed with a large number of sensor nodes. It may be designed to operate for months or years in hard to access areas. Thus, changing or replacing battery for such a large number of sensor nodes may not be practical or feasible. Thus, it is critical that WSNs operate as efficient as possible. The design of the network topology, protocols, routing algorithms, and even each individual sensor node is key to an efficient operation.

In single-hop based WSNs where sensor nodes are directly communicating with the controller or base station, the sleeping time mostly depends on the application. In multi-hop based WSNs, sink nodes act as the most active nodes in the network, so their batteries may drain out more quickly than other nodes.

3.2. E^2HUV Algorithm

A. Assumptions

Before describing the proposed algorithm, the following are the assumptions for the E^2HUV algorithm. These assumptions are based on the DEETP algorithm [9] for performance evaluation and comparison purposes:

- The time of multi-hop communication delay is negligible.
- Every sensor node including Rendezvous Points (RPs) has enough memory to store all data.
- The deployment of sensor nodes is uniform, so the UAVs know the location of nodes and RPs.
- In this scenario, the ad hoc WSN is in stable condition. The connections between nodes are stable and there is no redundant or isolated node.
- Every node has the same and fixed data communication range.
- Each sensor node will transmit one data packet; b bits in t time.

B. Notations

In this paper, a WSN is considered with a complete graph and modeled as G(V,E) where V is the set of sensor nodes, E is the set of edges between 2 nodes in V. When a sensor node i transmits b bits to node j, it will consume the amount of energy as follow [3][9]:

$$E_T\,(i, j) = b_{ij}\,.\,C_{ij}$$

Cij is the required energy when node i transmits one unit of data to node j. It is calculated by following formula:

$$C_{ij} = \propto + \beta . d^e{}_{ij}$$

$$Then: E_T\,(i, j) = b_{ij}\,(\propto + \beta . d^e{}_{ij})$$

Where $d_{i,j}$ is the Euclidean distance between 2 nodes i and j; $\propto$ is the energy consumption factor caused by the transmission; β is the energy consumption factor of the amplifier; e is the path loss which has range from 2 to 6 depending on the environment.

Similarly, the power consumption when node i receives b bits from node j is:

$$E_R\,(i, j) = b_{ij}\,.\,\gamma$$

Where γ is a given constant presenting the energy consumption per received bit.
So the total energy consumption per time unit at node i is:

$$ES = \sum_{j\in N} ET(i,j) + \sum_{j\in N} ER\,(i,j) = \sum_{j\in N} bij\,.\,Cij + \sum_{j\in N} bij\,.\,\gamma$$

The maximum length of the path that a UAV can cover, d, assuming a constant speed *v*:

$$d_{max} = P \cdot v$$

Where P is the maximum packet delay allowed in the network. ND(i) is the number of data packets that a sensor node i transmits to the closest RN:

$$ND(i) = G(i, T_j) + 1$$

Where G(i, Tj) is the number of sub nodes that node i has in the tree T. Based on the weight of a sensor node, the algorithm declares which node will become a RN. The weight of a sensor node i is calculated by the following formula:

$$Wi = ND(i).H(i, M)$$

Where Wi is the weight at node i; H(i, M) is the closest RN which is defined as:

$$H(i, M) = \{h_{i,mj} \mid \forall \boldsymbol{m}_k \in \boldsymbol{M},\ h_{i,mj} \leq h_{i,mk}\}$$

Where $h_{i,j}$ is the distance between node i and node j.

Standard Deviation (SD) is calculated to measure the imbalance of energy consumption among the different nodes. If SD is high, it indicates that some parts of the network tend to exhaust their energy quite rapidly compared to other parts [9].

$$SD = \sqrt{\frac{\sum_{i \in V}(ES[i] - \mu)^2}{|V|}}$$

Where ES[i] is the energy consumption by node i, *V* is the number of nodes, *u* is the average energy consumption.

The communication range of each sensor node is same over the whole simulation period, the flying altitude of a UAV needs to be carefully considered. If UAVs are acting as mobile sink nodes to collect the data from the sensor field and their altitude is properly calculated and adjusted, the network performance can be improved by reducing the connection range.

Notation	**Description**
G(V, E)	The network complete graph
V	The set of sensor nodes
E	The set of edges between 2 nodes
n	The total number of sensor nodes, n = |V|
v	The UAV's speed
d_{max}	Maximum tour length
$E_T(i, j)$	The energy consumption when node i send data packet to node j
$E_R(i, j)$	The energy consumption for receiving data packet
ES	The total energy consumption on a node. $(ES(i, j) = E_T(i, j) + E_R(i, j))$
$\propto$	The energy consumption factor caused by the transmission
β	The energy consumption factor of the amplifier
$d_{i,j}$	The Euclidean distance between 2 nodes *i* and *j*

e	The path-loss exponent (range from 2 to 6 depends on the environment)
SD	The standard deviation of the network
μ	The sensor nodes average energy consumption

TABLE 1: Summary of notations.

The flying height of the UAV is a function of the communication range of a sensor node. Figure 7 shows that the height h is the actual height of a UAV that will be maintained during the flight and while collecting data from the sensor field. R is the communication range of a node. The flying height of a UAV can be calculated as follows:

$$h = \sqrt{R^2 - \left(\frac{v.T'}{2}\right)^2}$$

Where *v* is the UAV speed; *T'* is the connection time.

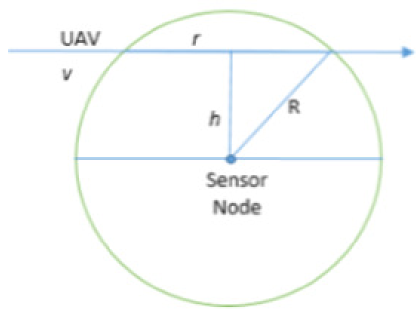

FIGURE 3: Height of a UAV.

	v = 1m/s
R = 20m	< 19.995m
R = 100m	< 99.861m

TABLE 2: UAV flying height under different speed.

C. E^2HUV Algorithm Description

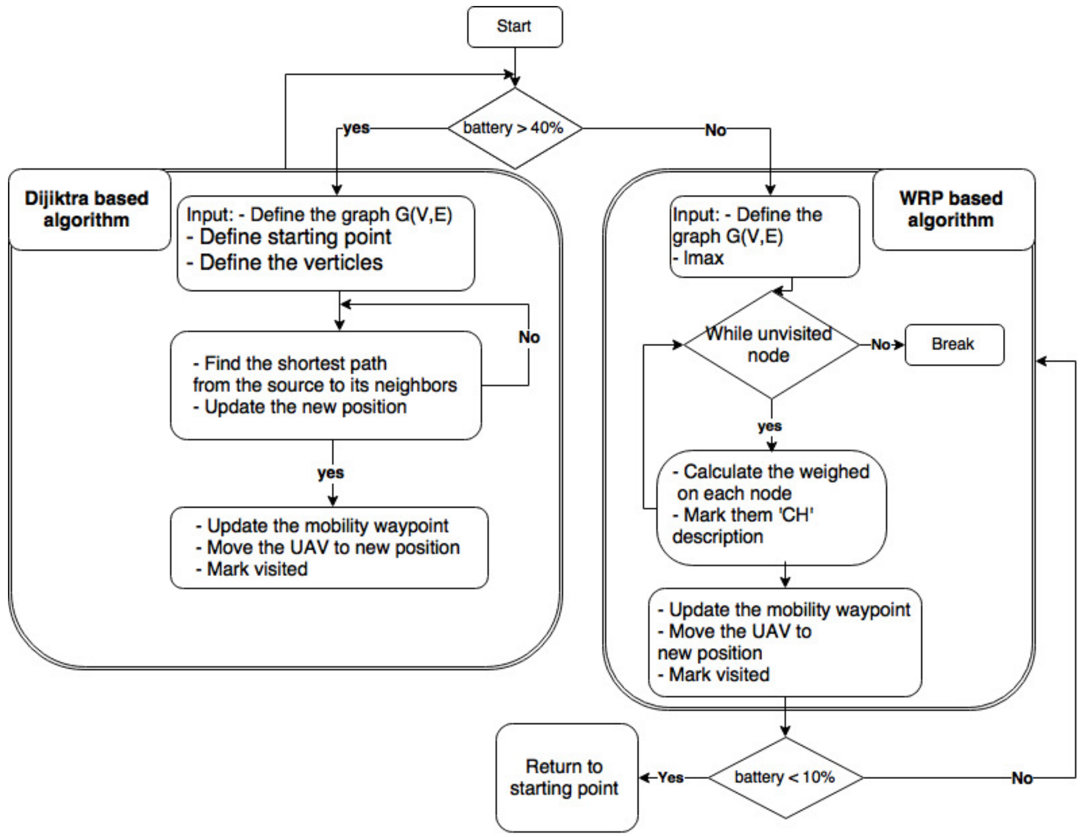

FIGURE 4: E^2HUV Flow Chart.

In this paper, the main objective of E^2HUV is to improve energy consumption and coverage in WSNs. The problem of limited-energy sink nodes is solved by using multiple UAVs. The UAVs offer better battery capacity that can be recharged and can cover larger areas. E^2HUV algorithm is based on Dijkstra's and Weight Rendezvous Planning (WRP). Figure 4 presents a flow chart of how the algorithm works. Depending on the UAV battery, the E^2HUV will accordingly operate and calculate the most appropriate path for the UAV to travel the sensor field. When the battery of a UAV is full or nearly full, the UAV will follow Dijkstra's algorithm. It will travel the shortest path. When the battery approaches a specified threshold, the proposed algorithm will function as a WRP-based algorithm. By selecting the RPs of the remaining unvisited sensor nodes, the UAV can collect data from RPs and forward to the base station. The ability to switch from Dijkstra's-based to WRP-based allows the mobile sink node (i.e. UAV) to achieve a balanced performance between coverage and lifetime.

Algorithm 1: E2HUV

```
Input Function G(N, E);
Input : Number of sensor nodes and UAVS num_nodes , num_UAV and
        startingpositing
int begin ;
all node == unvisited;
int remain_energy;
int threshold;
starting position (coordinate);
check energy_remain;
if(energy_remain > threshold)
Path = shortest path(Dijkstra); Basically, this step will find the closest neighbour of
  current node position UAV visit.
update UAV position; The update the UAV position to the new position of 16. the
  closest node.
node(i) == visited;
else if(energy_remain <= thresholdenergy_remain > drainout)
While(node(i) = unvisited)
G(node(i)) = WRP(node(i)); Will find the RPs by using Weigth Rendezvous Planning
  based algorithm.
update UAV position; The UAV will visit each and every RPs.
node(i) == visited;
else
update UAV to starting position; The UAV will go back when the battery goes down
  the warning level.
```

Algorithm 3: Dijkstra based algorithm

```
Dijkstra(G, w, s)
Input: G-Graph[V, E], w-weight, s-source
Output: M = (m0, m1, m2, mn) where M ∈ V
Begin
int visited = false;
for each Vertex ∈ V
dist[v] ← −1dist[s] ←
S ← emptyset
while Vertex[G] is not an empty set
u ← Extract_Min(Vertex[G])
S ← S ∪ u
for each edge (u,v) outgoing from u
if (dist[u] + weight(u,v) ¡ dist[v])
dist[v] ← dist[u] + weight(u, v)
visited[v] = true;
end;
```

Algorithm 2: WRP based algorithm

```
Input Function G(V, E) ;
Input : G(V, E), dmax
Output: M = (m0, m1, m2 , mn), where mi
  ∈ V \ D(Disthesetofnodesthatisalreadyvisited)
Begin
int Tn = 0;
int Wmax = 0, flag = 0, RP = -1;
float cost = 0;
position = X;
Boolean
mark = false, false, , removed = false, false,;
M = M ∪X;
Tn ++;
While Tn ≤| V \ D | do
Wmax = 0; flag = 0;
for i = 0 to V \Ddo
ND(i) = C(Tmi) + 1;
end
for i = 0 to V \D | do
if not mark(i) and (ND(i)∗H(i, M) > Wmax)then
RP = i; // assign node i is a RN
Wmax = ND(i)∗H(i, M);
flag = 1;
end;
end;
if !flag then break;
mark(RP) = true; M = M ∪RP;
Tn ++;
cost = distance(M);
if cost ≤
  dmaxthen//WhentheRNcandidatehasthecostfromthecurrentpo
for i = 0 to V \Ddo
if(C(Tmi) == 0, mark(i) == true, removed[i] == false, i ≠ RP)
then
removed[i] == true;
mark[i] == false;
M = M - i;
Tn − −;
end;
end;
end;
if(cost > dmax) then
M = M - R×P;
Tn − −;
Wmax = 0; flag = 0;
end;
end;
```

The complexity of the E^2HUV algorithm can be calculated as: $nlog(n^2)$. The complexity of Dijkstra's is *O(ElgV)* where there are *V* vertices in the graph; *E* is the total number of edges. The complexity of WRP is $O(n^5)$. Based on the complexity estimation, the E^2HUV algorithm has better performance compared to WRP.

Figure 5 shows an example of finding a travel path for a UAV using E^2HUV. The E^2HUV starts from the starting position near node 2. Consider node 2 as the first node that UAV will visit. Assume that the UAV battery is fully charged. The maximum tour length is d_{max} = 90m. A shortest

Path Tree (SPT) is created in Figure 5(a). When the algorithm starts, the UAV will visit node 2 first. From node 2, it finds the shortest edge from node 2 to its neighbors (12, 1, 8). In this case, it is node 12. As the operation continues, a path through nodes 2, 12, 11, 1 is created; as shown in Figure 5(b). When the battery reaches 40% (i.e. the set threshold in this case), the algorithm switches to WRP to find the closet set of RNs from its current position. Assume that the UAV is at node 1 when the battery level hits the threshold value. In its first iteration, node 10 is added to the path since it has the highest weight value among unvisited nodes. So M = [1, 10]. M is smaller than d_{max}, so node 10 is added to the tour. In the second iteration, node 7 becomes the closest RN and has the highest weight among the rest of nodes. M = [1, 10, 7] is smaller than d_{max}. So node 7 is added to the tour. The algorithm continues until the UAV has power enough to return to its base station.

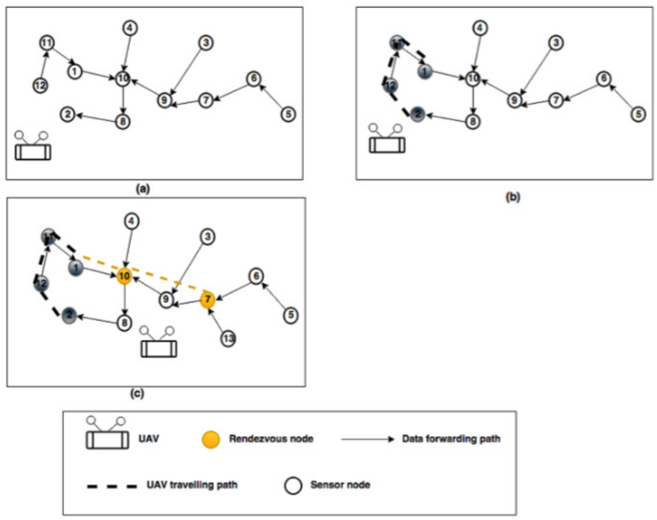

FIGURE 5: Example of E^2HUV with 13 nodes and one UAV.

3.3. Performance Analysis

3.3.1 Simulation Setup

The purpose of this simulation is to compare the E^2HUV algorithm with the two most commonly used algorithms; Dijkstra's and Weight Rendezvous Planning algorithms. Network Simulator 3 (NS3) is the simulator of choice with C++ programing language. The WSN consists of multiple sensor nodes deployed in a field of size 200 square meters; Figure 7. The simulation is run on 64-bit Ubuntu version 15.04.

FIGURE 6: Example of a UAV device - Parrot Bebop Drone. [10].

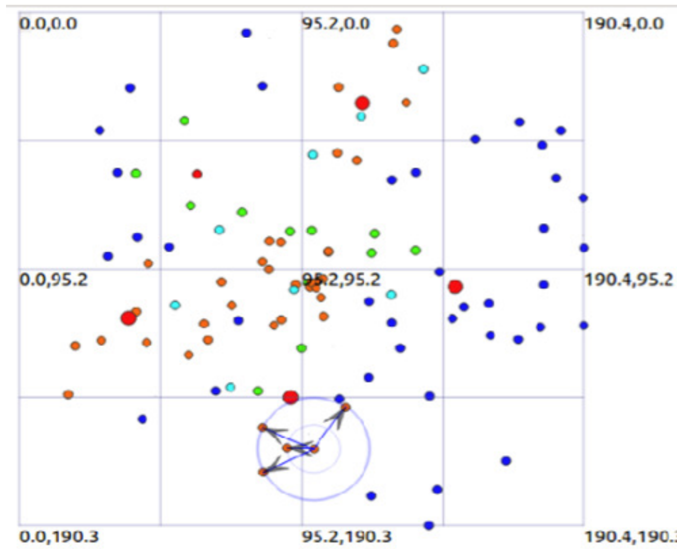

FIGURE 7: The E^2HUV Algorithm simulation.

The performance of the Dijkstra's, WRP, and E^2HUV algorithms are evaluated in this paper. All algorithms have been simulated under the same network and parameters conditions. In all the simulations, nodes 0, 2 and highest ID node are designated as mobile sinks (i.e. UAVs in this scenario). The results were averaged over 15 runs for each metric.

Parameter	Value
Number of sensor nodes	100
Number of UAVs	5
Maximum allowed packet delay	100 to 300 seconds
Sensor node transmission range	20m to 100m
UAV speed	1m/s
Packet length	210 bytes
Communication rate	448Kb/s
Energy consumption at transmitter	40mW
Energy consumption at receiver	20mW
Sensor node's battery	100J

TABLE 3: Simulation parameters.

3.3.2 Evaluation

In this implementation, the network size can vary between 2-200 sensor nodes. These sensor nodes are deployed in different clusters in the simulated area. There is a minimum of two UAVs, by default, to collect data from the sensing field. The number of UAVs is increased depending on the experiment scenario. For the purpose of implementation, it is assumed that all sensor nodes are fully charged and the speed of UAVs is constant at 1m/s. The simulation also follows the aforementioned assumptions.

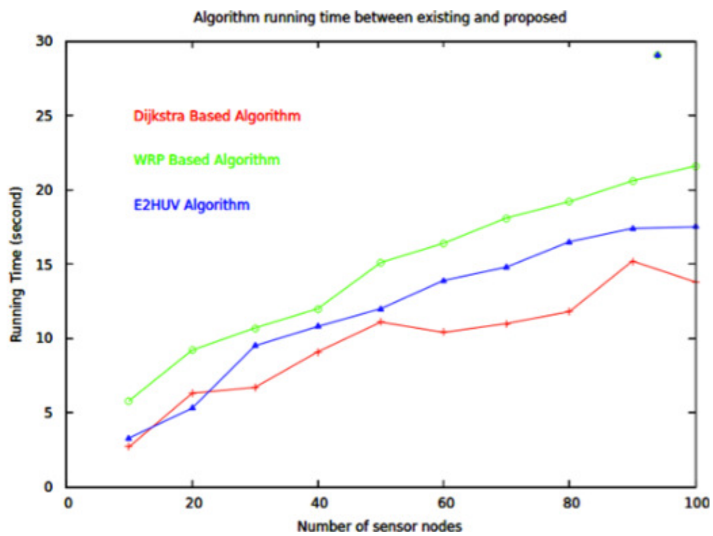

FIGURE 8: Algorithm running time for E^{2}HUV, Dijkstra's and WRP algorithms.

Figure 8 shows that E^{2}HUV algorithm outperforms WRP on running time. The Dijkstra's based algorithm seems to have the best running time among all algorithms. Dijkstra's based algorithm appears to take less time compared to WRP especially for large number of sensor nodes.

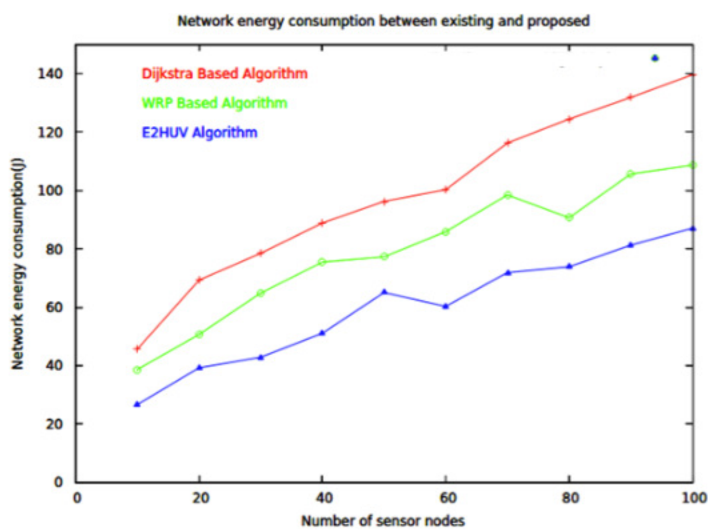

FIGURE 9: Network energy consumption for E^{2}HUV, Dijkstra's and WRP algorithms.

Figure 9 shows energy consumption performance for the three algorithms. The E^{2}HUV algorithm uses less energy approximately by 30% and 43% compared to WRP and Dijkstra's, respectively. It is worth noting that the proposed hybrid algorithm has greater running time than Dijkstra's but it costs less energy. The reason is that E^{2}HUV spends more time on computing and processing and less time on communications as compared to Dijkstra's. It is typical for such networks that communications would cost more energy than computing and processing.

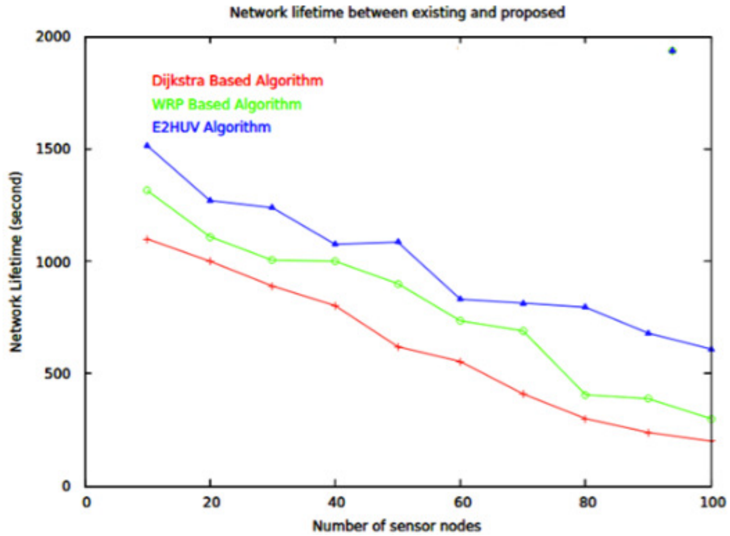

FIGURE 10: Network lifetime for E^2HUV, Dijkstra's and WRP algorithms.

As shown in Figure 10, the E^2HUV algorithm improves network lifetime by 11% and 20% as compared to WRP and Dijkstra's, respectively.

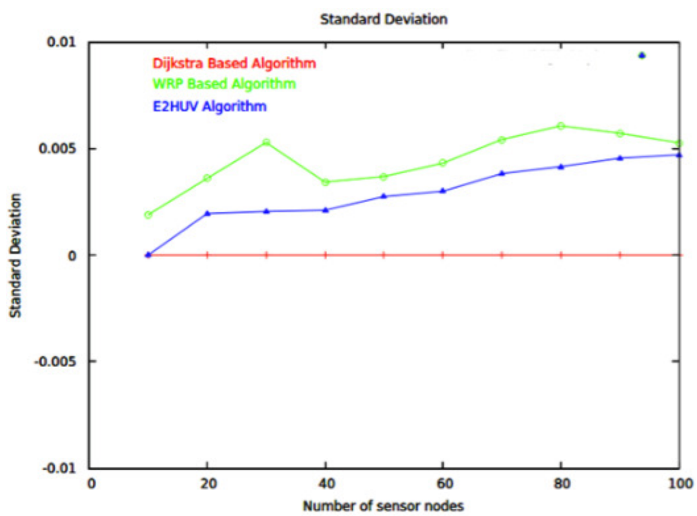

FIGURE 11: Standard deviation of sensor nodes' energy consumption.

Figure 11 shows energy consumption standard deviation (SD). The smaller the SD value is the longer the network lifetime is and the more uniform energy consumption is. E^2HUV tends to have a smaller SD compared to WRP allowing for better energy consumption balance among the sensor nodes. This can lead to a long network lifetime even with large number of sensor nodes. For SD, Dijkstra's serves as the lower bound with a value of 0.

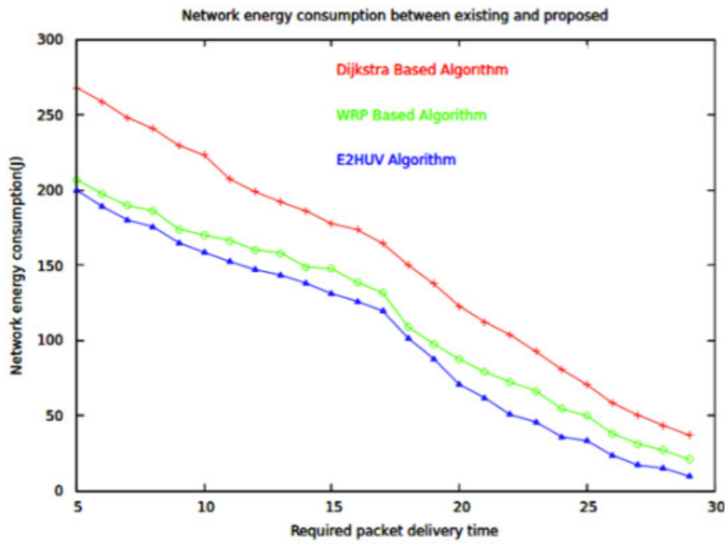

FIGURE 12: Network energy consumption for E^2HUV, Dijkstra's and WRP algorithms under different required delivery times for data packets.

Figure 12 shows energy consumption and network lifetime as a function of packet delivery time. The energy consumption for E^2HUV is reduced by 55% when the required packet delivery time is changed from 5 to 30 s. WRP and Dijkstra's observed a reduction of 20% and 12%, respectively.

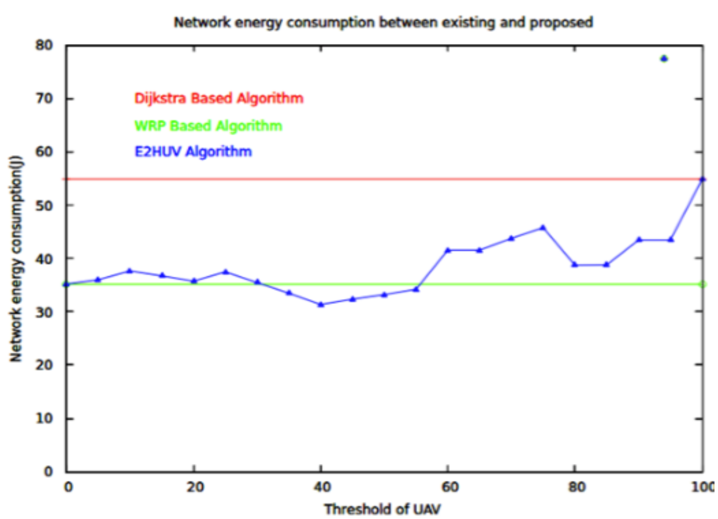

FIGURE 13: Network energy consumption with different UAV battery thresholds.

Figure 13 shows the performance of the proposed algorithm under different UAV battery thresholds from 0% to 100%. This is to observe the change in network energy consumption. When the battery threshold is very small, the proposed algorithm will operate as a WRP. When the threshold is nearly 100%, it will act as a Dijkstra's based algorithm. According to the results, the optimum energy consumption, for this scenario, is achieved when the threshold is at 40%.

4. ACKNOWLEDGMENTS

This work has been supported, in part, by National Science Foundation grant CNS-1042341.

5. REFERENCES

1. Salarian, Hamidreza, Kwan-Wu Chin, and Fazel Naghdy. An energy-efficient mobile-sink path selection strategy for wireless sensor networks. *Vehicular Technology; IEEE Transactions* 63.5 2014, 2407-2419.

2. Li, Jian, and Prasant Mohapatra. An analytical model for the energy hole problem in many-to-one sensor networks. IEEE Vehicular Technology Conference. Vol. 62. No. 4. IEEE 2005.

3. Sathyaraj, B. Moses, et al. Multiple UAVs path planning algorithms: a comparative study. Fuzzy Optimization and Decision Making 7.3 2008, 257-267.

4. Luo, Jun, and Jean-Pierre Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. INFOCOM 2005; 24th annual joint conference of the IEEE computer and communications societies; Proceedings IEEE. Vol. 3. IEEE 2005.

5. Sanchez, Erwing R., et al. An adaptive power-aware multi-hop routing algorithm for wireless sensor networks. Information Technology: New Generations (ITNG) 2011; Eighth International Conference on. IEEE 2011.

6. Gou, Haosong, and Younghwan Yoo. An energy balancing LEACH algorithm for wireless sensor networks. Information Technology: New Generations (ITNG) 2010; Seventh International Conference on. IEEE 2010.

7. Leng, Jinfu. Using a uav to effectively prolong wireless sensor network lifetime with wireless power transfer. Diss. University of Nebraska 2014.

8. Konstantopoulos, Charalampos, et al. A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks. Parallel and Distributed Systems; IEEE Transactions on 23.5 2012, 809-817.

9. Alomari, Abdullah, et al. A scheme for using closest rendezvous points and Mobile Elements for data gathering in wireless sensor networks. Wireless Days (WD) IFIP. IEEE 2014.

10. http://store.storeimages.cdn-apple.com/4839/as-images.apple.com/is/image

11. http://www.tkt.cs.tut.fi/kurssit/2456/wsnmotivation2.png

12. Cobano, Jose A.; Martinez-De Dios, J. R.; Conde, R.; Sánchez-Matamoros, J.M.; Aníbal, Ollero. Data retrieving from heterogeneous wireless sensor network nodes using UAVs. Journal of Intelligent And Robotic Systems 2010, 133-151.

13. Ho, Dac-Tu, Esten Ingar Grotli, and Tor Arne Johansen. Heuristic algorithm and cooperative relay for energy efficient data collection with a UAV and WSN. Computing, Management and Telecommunications (ComManTel) 2013; International Conference on. IEEE 2013.

14. Lawrence, Elaine, et al. Data collection, correlation and dissemination of medical sensor information in a WSN. Networking and Services 2009; ICNS 2009; Fifth International Conference on IEEE 2009.

15. Matin, Mohammad. Wireless sensor networks-technology and applications. Charalambos Sergiou, Vasos Vassiliou, Publisher: InTech 2012. ISBN-13 1304639176

16. Sotheara, Say, et al. Effective data gathering protocol in WSN-UAV employing priority-based contention window adjustment scheme. Globecom Workshops (GC Wkshps) 2014; IEEE 2014

17. Wang, Z. Maria, et al. Exploiting sink mobility for maximizing sensor networks lifetime. System Sciences 2005; HICSS 2005; 38th Annual Hawaii International Conference on. IEEE 2005.

18. Wei, Peng, Quanquan Gu, and Dengfeng Sun. Wireless sensor network data collection by connected cooperative UAVs. American Control Conference (ACC) 2013; IEEE 2013.

19. Xing, Guoliang, et al. Rendezvous design algorithms for wireless sensor networks with a mobile base station. Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing. ACM 2008.

20. Zuo, Yun, Zhihao Ling, and Yifeng Yuan. A hybrid multi-path routing algorithm for industrial wireless mesh networks. EURASIP Journal on Wireless Communications and Networking 2013, 1-12.