ESAC: An Energetic Sustainable Adaptive Clustering Protocol for Heterogenous Wireless Ad Hoc Networks (HANETs)

Nahid Ebrahimi Majd, Michael McDermott
Department of Computer Science and Information Systems
California State University San Marcos, United States
nmajd@csusm.edu, mcder008@cougars.csusm.edu

***Abstract*—We study adaptive clustering in heterogenous wireless ad hoc networks (HANETs) and propose ESAC, an energetic sustainable protocol that adaptively forms clusters using a weighted election function based on the current energy status of nodes in each region. The proposed protocol is distributed, and each node independently makes the decision to become a cluster head at the current round. The workload of a cluster head is heavy, including forming the cluster, collecting data from cluster members, data aggregation and transmission to faraway base station. ESAC selects an optimum number of cluster heads in each round among high-powered nodes that can afford the heavy duties of a cluster head. The protocol keeps track of the nodes' residual energies and selects the high-powered nodes as cluster heads more frequently than other nodes. As a result, although each node starts with a different initial energy, it adaptively tunes its energy consumption with its residual energy and the current energy status of other nodes, so that all nodes deplete their batteries at almost the same time. ESAC is energetic sustainable and the nodes last for as long as possible. Classical clustering protocols assume the same or certain number of initial energy levels for nodes, however ESAC is designed for HANETs where nodes have different energy levels. We validated the effectiveness and efficiency of our protocol through simulations. The analysis of our results showed that on an average ESAC reduces the instability into 46% and improves the total throughput by 42% and the stable phase throughput by 21% (and as high as 32%) in a HANET.**

Keywords*—*Heterogenous wireless ad hoc networks, Adaptive clustering, Energetic sustainability, Data fusion.

I. INTRODUCTION

Wireless ad hoc networks are self-organizing networks consisting autonomous nodes that form the network though ad hoc connections among nodes with no central coordination. Advances in IoT, sensor technology and low power electronics have presented a new type of ad hoc networks, called heterogenous wireless ad hoc networks (HANETs), which consist of different types of nodes, like wireless IoT devices, cellphones and cars, with different tasks, sensing capabilities and energy levels.

Two challenges of HANETs are (1) the energy usage of a node is typically high and dissimilar to other nodes depends on its application; (2) data collected by multiple nodes in the same region are corelated and might require a local data fusion before transmitted to the base station (BS).

We use an adaptive clustering technique to overcome these challenges. One of the most energy consuming duties of a node is data transmission to the remote BS. The required energy exponentially grows when the BS is further than a certain distance. We form a two-tier structure and divide the nodes into two layers: (1) cluster members, which associate with a cluster head and transmit their data to the BS through the head; (2) cluster heads, which collect data from their associated members, locally process the received data using data fusion techniques, and transmit the aggregated data to the remote BS. At each round, all nodes run the protocol, and a new group of nodes self-select to be the cluster heads. Then the remaining nodes associate with their closest cluster heads as the cluster members.

This hierarchical structure allows members to reduce their energy consumptions by short-distance transmissions to the local cluster instead of direct transmission to the far BS. On the other hand, data generated by multiple members are locally processed and aggregated at the cluster head and the cluster head sends only the aggregated data on behalf of the whole cluster to the BS. Although there is some energy dissipation at the cluster head to run the data fusion algorithm for all received signals from cluster members, the total energy dissipation is relatively much less than direct transmissions from members to the BS, and also the aggregated data is more accurate than individual members' signals. Data fusion enhances the common signal and removes the uncorrelated noises. The data fusion algorithm depends on the application. For instance, one of the algorithms used for acoustic signals is the beamforming algorithm.

A. Motivation

Serving as a cluster head is a heavy workload for a node, and to balance the workload among the nodes, the state-of-the-art adaptive clustering protocols utilize randomized rotation of cluster heads. However, this mechanism is unable to assign the correct amount of workload to the nodes, and in long term some nodes deplete their batteries much faster than other nodes. The reason is the dynamic nature of clusters, which change every new round.

Some of the main factors that impact the efficiency of this mechanism are: (1) The cluster heads that are further from the BS consume more energy to transmit data to the BS. Since energy required to transmit data exponentially grows by distance, the further nodes consume much more energy than the closer nodes and deplete their energies faster; (2) The members select their cluster heads based on the distance to the closest cluster head in the current round. If in a crowded region only one cluster head is selected at some round, a large number of nodes associate with this cluster head as its members, and it will consume a high amount of energy to receive data from all those members and run the data fusion algorithm for the bulky input data; (3) These mechanisms equally distribute the workload among nodes, which does not provide sustainability in a HANET where nodes are equipped with different energy levels.

These reasons motivate us to investigate an energetic sustainable adaptive clustering approach that distributes the workload based on the current energy status of nodes and results in a sustainable network with high throughput.

The remaining of this paper is organized as follows. Section 2 presents the related work. Section 3 presents the performance measures. Section 4 presents the proposed ESAC protocol. Section 5 presents the simulation setup and results. In Section 6, we conclude this paper.

II. RELATED WORK

An adaptive clustering protocol proposed in [1, 2], called LEACH, introduced a method for homogenous wireless ad hoc networks where nodes are initialized with the same energy level. LEACH uses a probability function that uniformly gives the same chance to all nodes to become a cluster head over a time period, called an epoch. Each node runs the heavy duties of a cluster head once in an epoch. Since all nodes become a cluster head once during this period, the energy dissipation is distributed among nodes.

The designers of LEACH protocol investigated the conditions where the total clustering, data fusion and transmissions from cluster heads to the BS consumes less energy than individual data transmissions from each node to the BS. They accurately determined the energy dissipation parameters for the transmitter and receiver such that this condition takes place. However, this protocol is unable to guarantee a uniform distribution of workload among nodes since it does not consider the dynamic size of clusters in adaptive clustering or different distances of the nodes to the BS. As a result, some nodes deplete their energies faster. When the energy level of a node falls below some threshold, the node stops working and dies. The node may still have very low amount of energy to keep sensing the environment and collect data, but since it is not strong enough to transmit the data, we consider it as a dead node.

SEP protocol proposed in [3], extended the LAECH protocol to a network of two groups of nodes: normal nodes with a lower energy level and advanced nodes with a higher energy level. They formulated the probability function for these two types of nodes and their results showed that their method is more efficient than LEACH in this scenario. ESEP protocol proposed in [4], extended SEP to three groups of nodes: normal, advanced and intermediate nodes with three energy levels.

B. *Functionality of LEACH protocol*

Each round in LEACH contains two phases: setup phase to form clusters, and steady state phase to transmit data. In the setup phase, clusters are formed; all nodes run the following probability function $T(s)$ and a few nodes self-select to be cluster heads. Then the cluster heads distribute HELLO messages to the network. All the remaining nodes receive these messages and reply to the first message they have received to associate with the closest cluster head.

$$T(s) = \begin{cases} \dfrac{P_{opt}}{1 - P_{opt} \cdot (r \bmod \frac{1}{P_{opt}})} & if\ s \epsilon G \\ 0 & otherwise \end{cases} \quad (1)$$

Each node becomes a cluster head once during an epoch of $1/P_{opt}$ rounds. P_{opt} is the desired percentage of nodes to become cluster heads in each round. The authors calculated the optimal value of P_{opt}, which depends on the number of nodes, the field size and the location of the BS. Each node selects a random number in the range $(0,1)$. If this value is less than probability $T(s)$ in the current round r, node s self-selects to be a cluster head in this round. G is the set of nodes that have not been selected as cluster heads in any previous round of the current epoch. If n is not in G, i.e. node n has already been selected as a cluster head in the current epoch, it cannot become a cluster head in the remining rounds of this epoch anymore. The function exponentially increases the probability to become a cluster head for the remaining nodes in the next rounds. In the last round of the epoch, the probabilities for all nodes that have not been selected as cluster heads become 1, and they will be selected. The whole process restarts in the next epoch.

At each round, once clusters are formed, the members transmit their data to their associated cluster heads using a TDMA protocol. The cluster head receives data from its cluster members, runs the data fusion algorithm and transmits the aggregated data to the BS.

The radio model of a node uses E_{elec} to run the transmitter/receiver circuitry. The energy dissipation to receive k bits data is $E_{RX}(k)$ and to transmit k bits data to a node in distance d is $E_{Tx}(k, d)$.

$$E_{RX}(k) = k \cdot E_{elec} \quad (2)$$

$$E_{Tx}(k, d) = \begin{cases} k \cdot E_{elec} + k \cdot E_{fs} \cdot d^2 & if\ d \leq d_0 \\ k \cdot E_{elec} + k \cdot E_{mp} \cdot d^4 & if\ d > d_0 \end{cases} \quad (3)$$

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}} \quad (4)$$

Distance d_0 is a threshold that distinguishes the two channel models: free space routing (fs) for short distance transmissions and multipath routing (mp) for long distance transmissions. The values of these parameters are given in Table 1. It is expected that the free space channel is used for transmissions between members and cluster heads and multipath channel is used for transmissions between cluster heads and the BS.

The designers of LEACH protocol accurately determined the value of E_{elec} such that the energy required to form the clusters, run the data fusion algorithm and transmit from cluster head to the BS is less than the total transmissions from individual members to the BS. The value of E_{elec} is relatively high, and as a result the receiver's workload is also relatively high. That is the reason why a large cluster results in a heavy workload for a cluster head.

Since each node is selected only once in an epoch, the heavy workload of cluster head is divided among nodes. However, energy dissipation of the nodes far from the BS is significantly higher than the closer nodes (in the order of d^4) since they use multipath channel. In a cluster, the further members from the head dissipate more energy than the closer ones. Also, a cluster head with larger number of members dissipates more energy for data reception and aggregation.

The abovementioned factors reduce the strength of LEACH. The protocol cannot dynamically adapt the energy dissipation according to the current situation of the nodes and clusters. These reasons motivated other LEACH-type schemes to improve the efficiency of LEACH. The proposed protocols in [5, 6] used multi-hop routing to transmit data to closer nodes to the BS and avoid long-distance transmissions. Reference [7] studied a multi-level LEACH to manage the energy dissipation in a multi-level structure. These protocols assume a homogenous initial energy for the network nodes. SEP [3] introduced a protocol for a heterogenous network of two energy levels. TSEP [8] proposed a threshold-sensitive protocol for a network of three energy levels. EA-LEACH [9] used residual energies to extend the network lifetime. **In our proposed protocol, the nodes are initialized to a variety of energy levels to implement a HANET.**

C. *Functionality of SEP protocol*

SEP assumes m percent of the nodes are advanced nodes and the rest are normal nodes. The initial energy of each normal node is E_0 and for advanced nodes is $E_0 \times (1+\alpha)$. SEP defines two different epoch lengths for the two types of nodes. Since the initial energy of an advanced node is higher, its epoch length is shorter, and it is more frequently selected as a cluster head. The aim is at each epoch the advanced nodes consume (1+α) times more energy than the normal nodes, so that all nodes deplete their energies at the same time.

The results showed that SEP made improvements over LEACH in a network with two energy levels. However, SEP also suffers from the same issues as LEACH since both these two protocols work based on the initial energies of nodes and are unaware of the current energy status of the nodes and clusters. Again, further cluster heads to the BS and the cluster heads with larger number of members deplete energies faster.

D. *Functionality of DEEC protocol*

DEEC [10] proposed a LEACH-type protocol to tackle the energy heterogeneities and prolong the network lifetime. DEEC estimates the network average energy ($estimated\ E_{avg}$) based on the round number and the total initial energy. Then uses E_s, the residual energy of node s to calculate P_s. Then uses the probability function $T(s)$ to make the decision whether node s will be a cluster head in the current round or not.

$$P_s = \frac{E_s}{estimated\ E_{avg}} \times P_{opt} \tag{5}$$

$$T(s) = \begin{cases} \frac{P_s}{1-P_s \cdot (r \bmod \frac{1}{P_s})} & if\ s \epsilon G \\ 0 & otherwise \end{cases} \tag{6}$$

G is the set of nodes eligible to be cluster heads at the current round r. DEEC improves the network lifetime. However, in this protocol, the epoch length for node s changes at each round. Node s does not keep track of its epochs or round numbers. If node s has not been a cluster head during the most recent $1/P_s$ rounds, it is eligible to be a cluster head. However, since the defined $T(s)$ function works with the current round r, there is no guarantee that node s becomes a cluster head at round $(1/P_s + 1)$ even if it has not been selected as a cluster head during the most recent $1/P_s$ rounds.

In LEACH and SEP, the $T(s)$ function is defined such that a node is selected as a cluster head once in an epoch but the defined $T(s)$ in DEEC does not guarantee that. Thus, the functionality of $T(s)$ in DEEC is not accurate, and the nodes will be selected less frequently as cluster heads. In long term, the nodes have less workload and longer lifetimes. However, DEEC cannot maintain the optimal number of cluster heads. Over the time, the average number of cluster heads declines from the optimal value even much earlier than the nodes start to die, and as a result, DEEC cannot reach a high throughput. Successors of DEEC [11, 12] used the same $T(s)$ function and studied three or more separate levels of initial energies and thresholds on energy levels or sensed data.

Our proposed ESAC protocol overcomes these problems with an accurate $T(s)$ probability function. Each node individually determines the correct length for its next epoch and tunes its frequency of being a cluster head so that in long term all nodes deplete their energies with an appropriate pace and while they maintain the throughput at the optimum level, they stay alive for a long time. As a result, ESAC provides high sustainability and throughput in both homogenous and heterogenous networks with no limitation on the nodes' initial energies.

III. PERFORMANCE MEASURES

We define the parameters we use to evaluate the performance of our protocol.

1. Stable phase: the time period from the start of network operation until the first node dies.
2. Unstable phase: the time period from the first death until the last node dies.
3. Network lifetime: the time period from the start of network operation until the last node dies.
4. Total throughput: total number of packets sent from cluster heads and received at the BS.
5. Stable phase throughput: total number of packets sent from cluster heads and received at the BS during the stable phase.

The network is **energetic sustainable** if its stable phase is long and unstable phase is short. It is desired to prolong the network lifetime, however when the nodes start to die, the network enters into an unstable phase, which declines the performance. When a node dies, no data could be collected from the region covered by that node, and the node cannot serve as a cluster head anymore either. The heavy workload of cluster heads is passed to the other nodes in the same area, and the energy depletion on those nodes occurs with a faster pace. The network enters into an unreliable and unpredictable status. To avoid this situation, we should tune the energy depletion on nodes such that all nodes remain alive as long as possible and all die in a short period of time.

If the unstable phase is long, it might not be possible or applicable to detect and replace the few dead nodes. Also, it might not be cost-effective to replace the entire network when a group of nodes remain alive much longer than the rest. If the network is energetic sustainable, we could estimate the stable phase, and it will be cost effective to replace the whole network at the end of stable phase since it is known that all nodes will die in a short period afterwards. This is the main strength of our ESAC protocol. Our results show that while ESAC provides very high energetic sustainability, it achieves higher throughput in comparison to the other protocols as well.

IV. THE PROPOSED ESAC PROTOCOL

In our ESAC protocol, the network is completely heterogenous and each node is initialized with a random energy level. However, unlike LEACH or SEP, our protocol does not use the initial energies of the nodes to distribute the load. Each node uses its own residual energy and the average energy of the network to self-select to be a cluster head. When members transmit their data packets to the cluster heads, they add their residual energies to the packet.

Operation	**Dissipated energy**
Transmitter/Receiver electronics (E_{elec})	50 nJ/bit
Data aggregation energy (E_{DA})	5 nJ/bit
Free space transmission power (E_{fs})	10 pJ/bit/m^2
multi path transmission power (E_{mp})	0.0013 pJ/bit/m^4

We could use one of the not-used header fields to store this information, e.g. the quality-of-service fields in the IP header. Once the cluster head receives all packets from members, it calculates the total residual energy of the cluster in the current round and sends it to the BS. At the end of round, the BS calculates the average energy of the network and broadcasts it to all nodes.

In our protocol, each node keeps track of its own epoch length and the round counter in its current epoch. We use r_s to represent the round counter for node s. When node s reaches the last round of its current epoch, it uses its residual energy E_s, and the most recent average energy of the network E_{avg} to calculate the length of its next epoch. Then the node resets both its round counter and G_s flag to 0. Once node s becomes a cluster head in an epoch, it sets its G_s flag to 1 for the current epoch, so that it will not be selected as a cluster head during the current epoch anymore. The length of next epoch for node s is $epoch_s$, and P_s is the weighted probability of node s.

$$P_s = \frac{E_s}{E_{avg}} \times P_{opt} \quad (7)$$

$$epoch_s = \frac{1}{P_s} \quad (8)$$

$$T(s) = \begin{cases} \frac{P_s}{1-P_s \cdot (r_s \bmod epoch_s)} & if\ G_s == 0 \\ 0 & otherwise \end{cases} \quad (9)$$

At each round, node s generates a random number. If it is less than $T(s)$, and node s has not been selected as a cluster head in any previous round in its current epoch yet, it will be selected as a cluster head. Round counter r_s starts from 0 and ends with $(epoch_s - 1)$ in the current epoch. The function $T(s)$ exponentially increases the probability to become a cluster head for node s. If the node is not selected to be a cluster head by the last round, the probability $T(s)$ becomes 1 at the last round, and the node will be selected. Then the node starts its new epoch. Unlike DEEC, our protocol guarantees that each node becomes a cluster head once in an epoch. Thus, in our protocol, the nodes are assigned the correct amount of workload to maintain the throughput on the optimal value. Node s is considered a dead node once its residual energy falls below a defined threshold $E_{deathThresh}$.

The weighted probability P_s assigns a short epoch length to high-energy nodes, thus those nodes will be selected as cluster heads more frequently and deplete their energies with a faster pace. However, some factors may cause a significant energy depletion on a node during an epoch, e.g. it is far from the BS, and data transmission to the BS was very energy-consuming; the number of its members was large when it served as a cluster head, so that the energy depletion for data reception and aggregation was high; it associated with relatively far cluster heads when it served as a member (note that E_{elec} is relatively high and even intra-cluster transmissions highly consume energy), etc. In such cases, the residual energy of the node falls during the epoch, and the probability function enlarges the next epoch length for node s, so that it will save energy while the stronger nodes run the heavy duties of cluster heads. Again, at the end of next epoch, the E_s/E_{avg} will be evaluated to determine the accurate length of the next epoch for node s. If the node could save energy during an epoch and became stronger than other nodes, its next epoch will be shorter, forcing the node to serve as a cluster head once in a short epoch.

V. SIMULATIONS AND RESULTS

We simulated and modeled the protocol with MATLAB. In our simulations, 100 nodes are uniformly distributed in a 100×100 field, and the BS is located at the center. For this configuration, P_{opt} is approximately 0.1 [3], meaning the optimal number of cluster heads at each round is 10% of the nodes. Table 1 presents the parameters used in the simulations. We evaluated the performance of our proposed ESAC protocol based on the following metrics.

1. Number of alive nodes per round.
2. Average number of cluster heads.
3. Total number of packets sent to the BS.
4. Stable phase length (in rounds).
5. Unstable phase length (in rounds).
6. Number of packets sent to BS during stable phase.

We evaluated the performance of our protocol in a heterogenous setting where m percent of nodes are advanced nodes with $E_0 \times (1+\alpha)$ initial energy, and the rest are normal nodes with a variety of random initial energies in the range $(E_0,\ E_0 \times (1+\alpha))$. We assume E_0 is 0.5 Joules. Fig. 1 shows the results for three heterogeneity parameter sets: $(m = 0.1, \alpha = 1)$, $(m = 0.2, \alpha = 1)$, $(m = 0.2, \alpha = 3)$. Fig. 2 illustrates a summary of results for three heterogeneity sets.

The results show that ESAC has the best energetic sustainability and the highest total and stable phase throughput. The first node dies later than LEACH and SEP and almost all nodes die in a short time afterwards. The lifetime of all nodes is almost the same, and as long as possible. The stable phase is long, and the unstable phase is short. ESAC prolongs the stable phase 42%, 40% and 86% over LEACH, and 30%, 30%, and 48% over SEP for the three scenarios respectively. Also, ESAC significantly reduces the unstable phase. On an average ESAC cuts the unstable phase into 24%, 24%, and 45% in comparison to LEACH, SEP and DEEC respectively. ESAC also improves the stable phase throughput by 54%, 32% and 21% over LEACH, SEP, and DEEC for the three scenarios respectively. As shown in Fig. 2, ESAC sends the highest number of packets to the BS.

In comparison to DEEC, the stable phase of ESAC is almost the same, however its unstable phase is much shorter and the total number of packets sent to the BS is significantly higher. On an average, ESAC sends 40% more data to the BS. The reason is as shown in Fig. 1, ESAC maintains the average number of cluster heads on the optimal value 10 nodes for almost the entire network lifetime, however DEEC starts with the optimal number or above and then constantly decreases the average number of heads. The data transmission rate falls early in the network lifetime even while the network is still in stable phase. DEEC constantly reduces the nodes' workloads during the network lifetime, and the nodes remain alive for a longer time. However, this reduction in workload results in a significantly less throughput in DEEC in all scenarios.

LEACH and SEP maintain the average number of cluster heads at the optimal value during the stable phase but once the network enters into its unstable phase, the number of cluster heads starts to fall, which decreases the throughput during the unstable phase. Since the unstable phase in LEACH and SEP is longer than ESAC (on an average 15 times longer), they keep sending low amount of data to the

BS for a long time. Therefore, the total amount of data sent to the BS is comparable to ESAC in the first two scenarios.

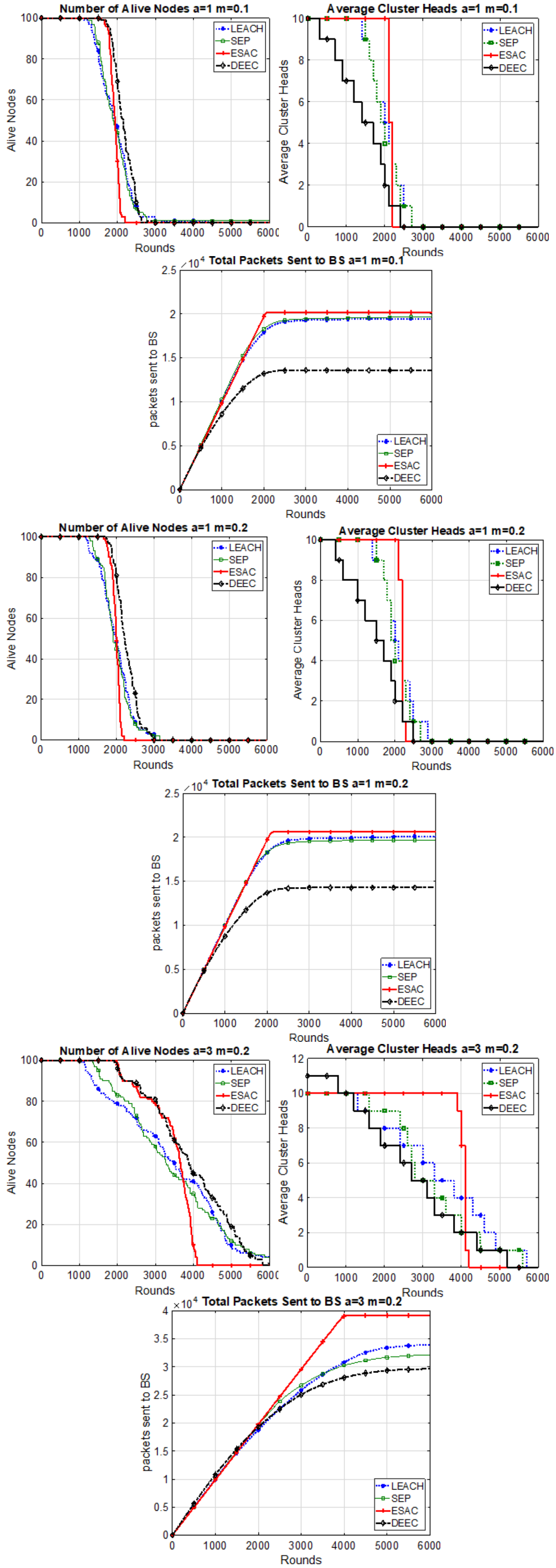

Fig. 1. Alive nodes, Cluster heads, and Packets sent to BS per round for 3 scenarios: $(m = 0.1, \alpha = 1)$, $(m = 0.2, \alpha = 3)$, $(m = 0.2, \alpha = 3)$.

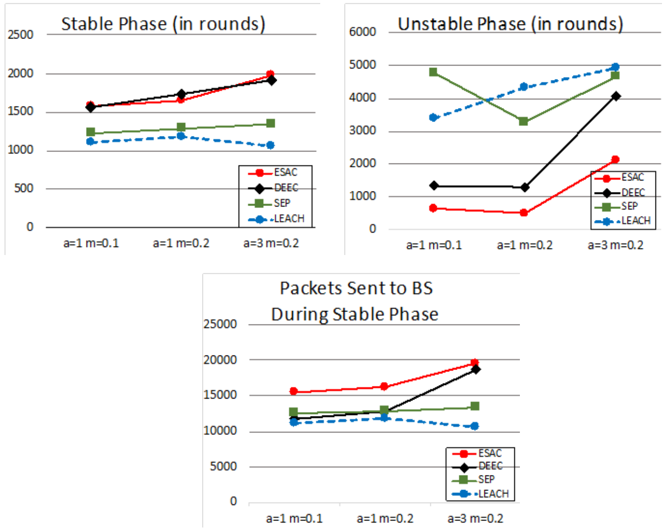

Fig. 2. The lengths of stable and unstable phases in rounds, and total number of packets sent to the BS during the stable phase for three scenarios: $(m = 0.1, \alpha = 1)$, $(m = 0.2, \alpha = 1)$, $(m = 0.2, \alpha = 3)$

ESAC has a long stable phase and maintains P_{opt} at the optimal value during the stable phase, which greatly improves the stable phase throughput. The total number of packets sent to the BS by the time the first node dies is the highest in ESAC in all scenarios. It is important to consider that when the network enters into unstable phase, the data reaches the BS is not accurate and should not be used to evaluate the performance. ESAC improves the total number of packets sent to the BS during the stable phase in all scenarios.

ESAC improves the number of packets sent to the BS during the stable phase by 32% and 26% over DEEC for the first two scenarios and 5% in the third scenario. Referring to the average number of cluster heads in Fig. 1, when the difference between energy levels of the two types of nodes is very high ($\alpha = 3$ where advanced nodes initiate with 4 times more energy than average energy of normal nodes), the advanced nodes are selected very frequently as cluster heads. On an average, DEEC selects 11 cluster heads for almost half of rounds in its stable phase and then this number declines to 7 by the time the first node dies. Unlike ESAC, DEEC does not maintain consistency in throughput over time, which results in low stable phase throughput. However, in this scenario the overall stable phase throughput gets a large number since DEEC selects a very high number of cluster heads at the beginning of its lifetime. We will discuss this scenario with more details in this section.

Overall, ESAC provides the best energetic sustainability and stable phase throughput in all scenarios. ESAC maintains consistency in the average number of cluster heads over time and keeps that on the optimal value while in LEACH and SEP, this value starts to fall when the first nodes die, and in DEEC this value constantly decreases.

The unstable phase of ESAC for the last scenario where ($m = 0.2, \alpha$=3) is relatively long, which is not expected. We calculated the energy consumption at clusters, and the calculations proved that with the given energy dissipations parameters for transmission/reception, it is impossible to get an optimal result in this scenario. The proof is as follows.

We define $epoch$ as the length of epoch for a normal node. During an $epoch$, a normal node serves once as a cluster head and $(epoch - 1)$ times as a member. We assume in an optimal scenario an advanced node serves β times as a cluster head and $(epoch - \beta)$ times as a member. During an $epoch$, the total energy consumed by the set of advanced nodes must be $(1 + \alpha)$ times the total energy consumed by the set of normal nodes. The initial energy of an advanced node is $(1 + \alpha)$ times the average initial energy of a normal node. There are n nodes in the network. E_{CH} is the average energy required to serve as a cluster head, and E_{mem} is the average energy required to serve as a member in a round. Thus, the following equation must be true to get the optimal result where normal and advanced nodes will deplete their energies at the same time.

$$n \cdot (1 - m)(E_{CH} + (epoch - 1) \cdot E_{mem})(1 + \alpha) =$$
$$n \cdot m \cdot (\beta \cdot E_{CH} + (epoch - \beta) \cdot E_{mem}) \quad (10)$$

The network is located in an area of $A = 2a \times 2a$ square meters, where $a = 50$ in our experiment. The E_{CH} and E_{mem} are calculated as follows [2, 3].

$$E_{CH} = L \cdot E_{elec} \cdot \left(\frac{1}{p_{opt}} - 1\right) + L \cdot E_{DA} \cdot \frac{1}{p_{opt}} + L \cdot E_{elec} + L \cdot E_{fs} \cdot d_{toBS}^2 \quad (11)$$

$$E_{mem} = L \cdot E_{elec} + L \cdot E_{fs} \cdot d_{toCH}^2 \quad (12)$$

where L is the packet size, and d_{toBS} is the average distance between a cluster head and the BS and d_{toCH} is the average distance between a member and the cluster head.

$$E[d_{toBS}] = \int_{-a}^{a} \int_{-a}^{a} \sqrt{x_i^2 + y_i^2} \left(\frac{1}{4a^2}\right) dx_i y_i = 0.765\ a \quad (13)$$

$$E[d_{toCH}^2] = \iint (x^2 + y^2)\, \rho(r, \theta) dx dy = \frac{(2a)^2}{2 \cdot \pi \cdot n \cdot p_{opt}} \quad (14)$$

If we assume the epoch length is $epoch = \frac{1}{P_{adv}} = 10$, for the given parameters at table 1, β will be almost 16, meaning during an epoch of 10 rounds, an advanced node should become a cluster head for almost 16 times, which is impossible since a node can be selected as a cluster head at most once in a round. Thus, with $(m = 0.2, \alpha = 3)$, it is impossible to reach the optimal energetic sustainability. This is true for any scenario that a small portion of nodes are initiated with a significantly more energy than the average energy of the remaining nodes. The main reason is the members also consume high energy (almost 1/11 of a cluster head with the given parameters). SEP aims that during a normal epoch each normal node uses $(1/(1 + \alpha))$ times the energy used by an advanced node. However, when α is large, since a normal node frequently serves as a member, it consumes slightly more than that at each round. In long term, normal nodes, deplete their energies faster than expected, and advanced nodes remain alive much longer than normal nodes.

In our proposed ESAC protocol, we consider the residual energies of nodes at each round to select the cluster heads. However, although ESAC provides the optimal epoch length for each node, the calculated lengths are decimal point numbers and should be rounded to be used as the length of an epoch in terms of number of rounds. This unremovable rounding reduces the accuracy of method and results in a relatively long unstable phase in this scenario.

However, the heterogeneity of initial energies is extremely high in this scenario, which is typically not the case in real scenarios. In other scenarios where the heterogeneity is more realistic, it is possible to get the optimal result with very high accuracy, and ESAC presents very good results.

Although we proved it is impossible to get an optimal result in the last scenario where $(m = 0.2, \alpha = 3)$, the ESAC protocol presents high sustainability and stable phase throughput in this scenario as well.

VI. CONCLUSIONS

In this paper we introduced ESAC, a distributed adaptive clustering protocol for HANETs. Our protocol distributes the workload among nodes in an appropriate pace based on the current energy status of the nodes in the network. Although the nodes start with different energy levels and consume different amounts of energy during the network lifetime, the network remains stable for as long as possible and all nodes die at almost the same time. ESAC is a robust protocol that provides very high energetic sustainability and stable phase throughput.

The proposed protocol can be used as the basic energetic sustainable adaptive clustering protocol for next research studies including multi-hop and hierarchical adaptive clustering for both homogenous and heterogenous ad hoc networks.

VII. REFERENCES

[1] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocols for Wireless Microsensor Networks,” in Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS-33), January 2000.

[2] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660670, October 2002

[3] G. Smaragdakis, I. Matta, A. Bestavros,“SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks,” In Proceedings of 2nd International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004), 2004.

[4] F. A. Aderohunmu and J. D. Deng,“An Enhanced Stable Election Protocol (SEP) for Clustered Heterogeneous WSN,” Department of Information Science, University of Otago, New Zealand, 2009.

[5] M. O. Farooq, A. B. Dogar, and G. A. Shah, “MR-LEACH: multi-hop routing with low energy adaptive clustering hierarchy,” Fourth IEEE International Conference on Sensor Technologies and Applications, 2010.

[6] A. Razaque, M. Abdulgader, C. Joshi, F. Amsaad, M. Chauhan, “P-LEACH: energy efficient routing protocol for wireless sensor networks,” IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1-5, 2016.

[7] R. K. Kodali and K. A. Naveen, “Multi-level LEACH protocol model using NS-3,” IEEE International Advance Computing Conference (IACC), 2014.

[8] A. Kashaf, N. Javaid, Z. A. Khan, and A. K. Imran, “TSEP: Threshold-Sensitive Stable Election Protocol for WSNs,” IEEE International conference of Frontiers of Information Technology, 2012.

[9] N. Ebrahimi Majd , P. Chhabria, A. Tomar, “Energy Aware LEACH Protocol for WSNs,” In Advances in Information and Communication. FICC 2019. Lecture Notes in Networks and Systems, vol 70. Springer.

[10] L. Qing, Q. Zhu, and M. Wang, “Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks,” Computer communications, no. 29, vol. 12, 2006

[11] B. Elbhiri, S. Rachid, and D. Aboutajdine, “Developed Distributed Energy-Efficient Clustering (DDEEC) for heterogeneous wireless sensor networks,” 5th IEEE International Symposium On I/V Communications and Mobile Network, 2010.

[12] G. K. Reddy and L. N. Devi, “A Review on Clustering Protocols with Energy heterogeneity in Wireless Sensor Networks,” IEEE International Conference on Communication, Computing and Internet of Things (IC3IoT), 2018.