Received June 25, 2019, accepted July 31, 2019, date of publication August 2, 2019, date of current version August 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932951

A Congestion-Aware Clustering and Routing (CCR) Protocol for Mitigating Congestion in WSN

MOHAMMED FARSI[1], MAHMOUD BADAWY[2], MONA MOUSTAFA[2], HESHAM ARAFAT ALI[2], AND YOUSRY ABDULAZEEM[3]

[1]College of Computer Science and Engineering, Taibah University, Yanbu 30012, Saudi Arabia
[2]Department of Computer Engineering and Systems, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[3]Computer Engineering Department, Misr Higher Institute for Engineering and Technology, Mansoura 35516, Egypt

Corresponding author: Mahmoud Badawy (engbadawy@mans.edu.eg)

ABSTRACT Wireless sensor networks (WSN) have been investigated as a powerful distributed sensing application to enhance the efficiency of embedded systems and wireless networking capabilities. Although WSN has offered unique opportunities to set the foundation for using ubiquitous and pervasive computing, it suffered from several issues and challenges such as frequently changing network topology and congestion issue which affect not only network bandwidth usage but also performance. The main objective of this study is to introduce a congestion-aware clustering and routing (CCR) protocol to alleviate the congestion issue over the network. The CCR protocol is proposed to decrease end-to-end delay time and prolong the network lifetime through choosing the suitable primary cluster head (PCH) and the secondary cluster head (SCH). The experimental results demonstrate that the effectiveness of the CCR protocol to satisfy the quality of service (QoS) requirements in increasing the network lifetime and raising the number of packets sent alike. Moreover, the CCR outperforms other state-of-the-art techniques in decreasing the overflow of data, and thus the network bandwidth usage is reduced.

INDEX TERMS Congestion control, clustering protocols, pervasive computing, quality of service (QoS), routing protocols, ubiquitous computing, wireless sensor network (WSN).

I. INTRODUCTION

Wireless Sensor Network (WSN) affects everyday life, as a seed of smart applications and pervasive systems alike. WSN is equipped with the traditional WiFi antennas and a group of internet connected-devices otherwise called ''smart nodes'', which can sense and record the natural phenomena and physical conditions from their surrounding environments such as humidity, temperature, pressure, and pollution level [5]. Due to its flexibility and its communication capability it is possible to utilize the real information generated by agent devices through a virtual layer. Thus, reliable transmission is provided [5], [9]. By using WSN a set of promising solutions to ensure the sustainability of smart applications such as Internet of Things (IoT) applications and military applications have been demonstrated. Figure 1 illustrates several smart applications use WSN as a core of their work.

The associate editor coordinating the review of this manuscript and approving it for publication was Sara Pizzi.

Conventional WSN involves hundreds or even thousands smart nodes can be deployed in a dynamic manner such as Mobile Ad Hoc Networks (MANETs) [8] and wireless ad hoc network [7] or in a static manner such as two-dimensional mesh and n-dimensional mesh network [1]. These sensors are divided into a set of wireless sensor nodes and a data center or sink node [5]. Three main operations are introduced by these nodes: (i) data collector, (ii) data processing, and (iii) data transmission. Besides, several functional modules to manage them such as a sensing unit, a processing unit, a storage unit, a transceiver unit, a power unit and, a power generator. The data streamed from the wireless sensor nodes travels to the data center node or sink node via wireless channel [5], [9].

Although, the investment of WSN has become the inevitable corollary of the developing countries with less-developed infrastructure due to its low cost and its communication capability [3]. However, the limitations of network bandwidth as well as increasing data loss rate and collision are still big challenges in WSN [2], [10]. Hence,

FIGURE 1. A set of smart applications which dependent on WSN.

satisfying the Quality of Service (QoS) demands through offering an efficient routing protocol based on clustering technique can reduce the network congestion issue and enhance the overall network performance [2].

The main objective of this study is to introduce a Congestion-aware Clustering and Routing (CCR) Protocol for better network performance in terms of throughput, end-to-end delay time, delivery ratio, and network lifetime. Surly, to achieve these objectives, many challenges and issues should be considered such as battery dependency, storage unit size, and sending data to the same receiver node many times. The contributions of the proposed CCR Protocol are as follows:

- Low Overhead: decreasing the overhead of performing the setup phase of each round, because the setup phase is done only once in the first round and used to divide the network area into levels and sectors to create clusters with an equal number of nodes.
- Load Distribution: distributing the load of cluster head node (CH) role between all nodes, as at the beginning of each round, the roles of primary cluster head (PCH) and secondary cluster head (SCH) rotate between all nodes in the cluster.
- Stability: using SCH to help PCH to do its role in the transmission of data, and the possibility of adding new nodes in any round and in any cluster.
- Reliability: choosing PCH and SCH based on specific performance metrics.
- Scalability: the possibility of adding new nodes in any round.
- Fault-tolerance: using fault tolerance methods to increase the packet delivery ratio.

The rest of this paper is organized as follows: the next section provides a brief background of WSN. In section 3, the related work is reviewed. In section 4, the proposed CCR protocol is presented. In section 5, the numerical results with the case study to validate the feasibility of the proposed protocol are analyzed. In section 6, the future work is demonstrated and the paper is concluded.

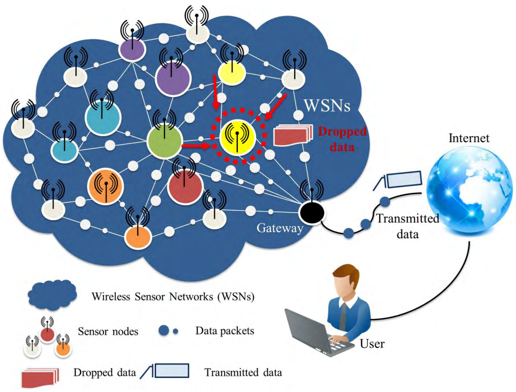


FIGURE 2. Congestion and packet loss problem.

II. BACKGROUND

During the last years, WSN has gained significant attention in both academic research and IT industry field due to its flexibility and its communication capability; especially when traditional networks such as Long-Term Evolution (LTE) or Terrestrial Trunked Radio(TETRA) are not operational [4]. However, WSN faces several profound issues and challenges may adversely affect its potential performance such as frequently changing network topology, longer end-to-end delay, route coupling, and high packet loss. Hence, providing the innovative solutions to reduce the congestion issue over the network has become necessary not only to decrease network bandwidth and power usage but also to prolong the network lifetime as much as possible [6]. In this section, the fundamental concept of congestion control is discussed as well as demonstrating, in brief, the common solving solutions to this problem and the congestion issue in network layer is also reviewed.

A. CONGESTION PROBLEM

When several sensor nodes send data to a single sink node at the same time, there is a big chance of the congestion in the network. The main reasons are, among others, rather limited availability of bandwidth and a finite network capacity [6]. This phenomenon is illustrated in Figure 2.

There are two main reasons for the congestion occurrence in WSN, the poor possibilities of nodes and the nature of the wireless channel. First, congestion in WSN happens in nodes because of the limited memory, the slow processor and the limited energy of nodes. Second, congestion in WSN happens in the network due to the network nature, event-driven nature, channel interference,and reporting rate. So, protocols designed for WSNs must be lightweight and scalable to extend the lifetime of the network to the maximum.

B. CONGESTION CONTROL

A congestion control mechanism has three main components: congestion detection, congestion notification, and

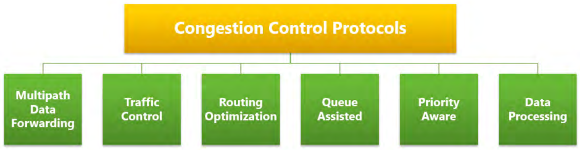


FIGURE 3. Classification of congestion control protocols.

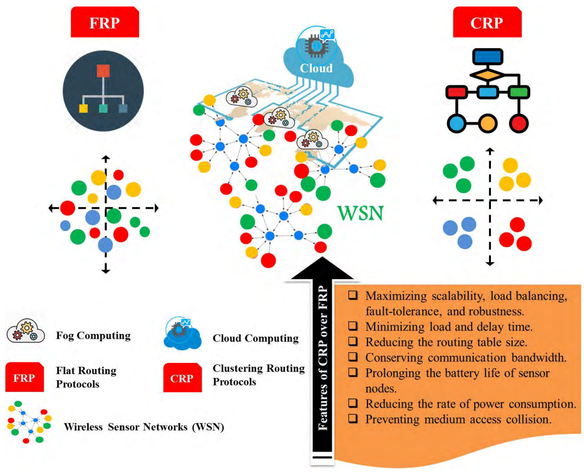


FIGURE 4. Features of CRP over FRP.

congestion control. There are several research efforts on congestion control in WSN [12]. A congestion control algorithm can be designed based on either congestion avoidance or congestion control [13]. Congestion control protocols can be classified into six main categories as shown in Figure 3. This paper proposes a clustering-based congestion control in the network layer by introducing a routing strategy for mitigating congestion.

C. CONGESTION CONTROL IN THE NETWORK LAYER

Routing protocols in WSNs can be divided into two main categories: (i) Flat Routing Protocols (FRP) and (ii) Clustering Routing Protocols (CRP). FRP distributes routing information between routers that are connected without any organization or segmentation structure. It enables the delivery of packets among routers through any available path without considering network hierarchy, distribution, and composition. In CRP, each cluster consists of the CH and other member nodes (MNs). The CHs are responsible for data aggregation, information dissemination, and network management. The MNs are responsible for identifying events and collecting information in their surroundings. CRP has a variety of benefits compared with FRP, as shown in Figure 4.

There are some critical considerations in the design process of clustering protocols for WSNs:

- Cluster formation: select the best possible clusters and CHs with a low number of exchanging messages and low total time complexity.
- Application dependency: a variety of applications can use this designed protocol.
- Secure communication: an essential point, especially in military applications.
- Synchronization: use the suitable MAC protocol to achieve Synchronization (e.g. slotted transmission schemes such as TDMA).
- Data aggregation: optimized according to specific application requirements.

III. RELATED WORK

As shown in Figure 5, hierarchical or clustering routing protocols can be divided into five main categories as follows: block-based, chain-based, grid-based, tree-based and area-based. The following subsections review the most recently proposed popular clustering protocols in the literature.

A. BLOCK-BASED CLUSTERING ROUTING PROTOCOLS

Low-Energy Adaptive Clustering Hierarchy protocol (LEACH) [14] is a self-organized, adaptive clustering protocol. The operation of LEACH is broken up into lots of rounds. In each round, nodes organize themselves into clusters. Each cluster contains only one CH node and many of MNs nodes, where CH node receives data from MNs nodes and performs signal processing functions on these data then sends these aggregated data to the BS. LEACH had many benefits such as balancing energy consumption, using a TDMA on MAC, aggregating data from CH nodes that lead to limiting the high amount of traffic and saving energy. It can also add new nodes or remove dead nodes in each round. However, it suffers from drawbacks such as the random selection of CHs, residual energy that is not considered in the selection of CH, single-hop inter-cluster routing that leads to energy consumption in large-regional networks and dynamic cluster that causes extra overhead. Table 1 lists the most recent successors of LEACH protocol, showing their technique, advantages, and drawbacks.

B. A CHAIN-BASED CLUSTERING ROUTING PROTOCOL

Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN) [32] is the first protocol developed for reactive networks and it combines the hierarchical technique with the data-centric approach. First, the formulation of clusters and their cluster heads is done. Then, cluster heads broadcast two critical values to their members, Hard-Threshold (HT), and Soft-Threshold (ST). HT is the absolute value of the sensed attribute. As if sensing the value of any node reaches the HT value, it must transmit this value to its CH. ST is the amount of change of sensed attribute, which enables the node to send the sensed value. TEEN has advantages such as (i) using HT makes the node transmit only when the sensed attribute is in the range of interest, (ii) using ST makes the node transmit when there is an amount of change in the sensed value, (iii) reducing the number of transmissions and energy and (iv) being suitable for time-critical applications. However, the drawbacks are found when identifying the value of any node does not reach HT or ST. In that case, this node does not send any value. This makes the BS unable to know

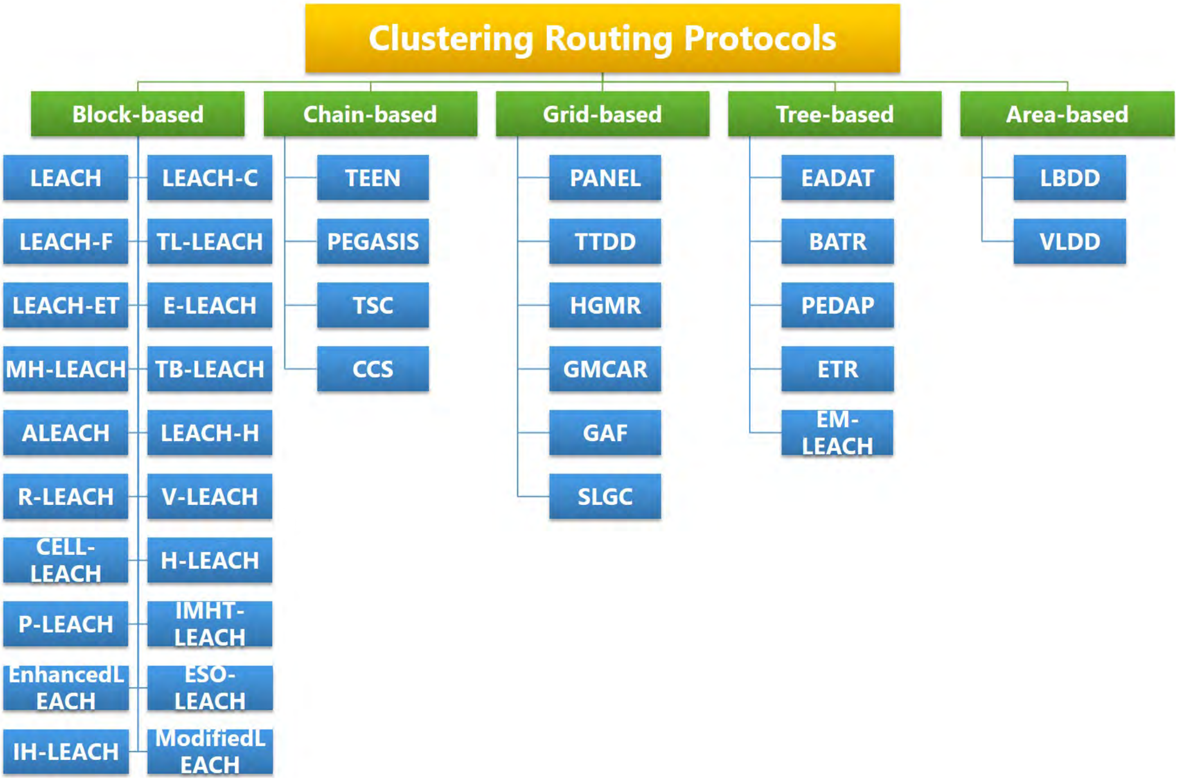


FIGURE 5. Taxonomy of clustering routing protocols.

if this node is alive or dead, and unable to know if the CHs are not in the range of communicating with each other, and thus the data may be lost.

Power Efficient Gathering in Sensor Information System Protocol (PEGASIS) [33] uses a greedy algorithm to organize the network sensors to form a chain, starting from the farthest node, to ensure that the node away from the sink has close neighbors. The BS computes this chain and broadcasts it to all the nodes. In each round, only one node takes the role to be the leader in transmitting to the BS. This role is rotated between all nodes in the network except among nodes with relatively distant neighbors along the chain. This chain guarantees that any node can receive from and transmit to close neighbors. This makes the distant transmissions as small as possible. Advantages of PEGASIS are unifying consumption of energy between all nodes; reducing overhead by using the chain instead of forming dynamic clusters and decreasing the amount of data transferred. Disadvantages of PEGASIS are that all nodes can communicate directly with the BS, they suffer from delays, they are not scalable as all nodes must have global knowledge of the network to run the greedy algorithm and they are not suitable for time-varying topologies.

C. A GRID-BASED CLUSTERING ROUTING PROTOCOL

Position-based Aggregator Node Election Protocol In (PANEL) [34] partitioned the network area is into geographical clusters before the deployment of the network. Each sensor node is aware of its geographical position *P*, and distinguish geographical information about its cluster by knowing the coordinates of the lower left corner of its cluster. The operation of PANEL is divided into epochs. In each epoch, computing reference points in each cluster is done first, and then these reference points are used to compute the aggregator points. In PANEL, each node acts as an aggregator node in equal chances, so it ensures the load balancing and saves energy, and it supports the asynchronous application. The drawbacks of the PANEL are that it cannot be applied to the dynamic WSN applications because clusters are predetermined before deployment, and it uses special hardware and software like the GPS to find the geographical position of the nodes.

D. AREA-BASED CLUSTERING ROUTING PROTOCOLS

Line-based Data Dissemination (LBDD) [35] defines a vertical strip or line of nodes, which divide the field of deployment into two equal portions. The nodes on this strip or line are referred to as inline nodes. This line acts as a gathering region for data storage. Sensor data are sent to the line and the first in-line node encountered stores the data. The sink sends a data query to the line and the query is propagated through the line until the in-line node storing the data is reached. The in-line node then forwards the data directly to the sink, and data dissemination is completed. It assumes that each node knows

TABLE 1. Comparison of recent LEACH successors protocols.

Protocol	Technique	Advantages	Drawbacks
LEACH-C [14]	uses a central control algorithm to form clusters	Improve performance by 20% to 40%	every node require GPS which is a costly device and it consumes extra energy
LEACH-F [15]	Run setup phase in the first round	No setup overhead	Consumes much power, cannot add new or remove dead nodes
TL-LEACH [16]	Uses two level of transmission	Reduce power consumption	Increases delay time
LEACH-ET [17]	Uses energy threshold to increase the proportion of steady state in each round	Energy efficiency enhanced	Consumes energy in transmitting control messages
E-LEACH [18]	Select CH according to the amount of residual energy	Increase network lifetime by 40%	Fixed time round leads to the waste of energy
MH-LEACH [18]	Provides inter and intra-multi-hop communication to send the data	Minimize energy consumption especially for large size network	Suffers from hot spots and limited scalability
TB-LEACH [19]	Fixed number of selected CHs each round based on the random time interval	Improves network lifetime by 20% to 30%	Not suitable for large scale networks
A-LEACH [20]	CH is the node with more energy, data is processed by mobile agents	Improves the reliability of data transfer with minimum node failure probability	Not suitable for large scale networks
LEACH-H [21]	Uses the location and energy to select the optimal CH	It combines the centralized protocol with the distributed one	Suffers from large overhead due to the selection of a new CHs list by CH
R-LEACH [22]	Uses position information and node density to improve the cluster structure	Provides multihop inter-cluster routing	Not suitable for large scale networks
V-LEACH [18]	Solves the problem of CH node death by adding vice CH which takes the role of dead CH	No loss in data, increases the network lifetime	Increases the overhead at the setup phase
CELL-LEACH [23]	Divides the network into cells and every 7 cells form one cluster with one CH	Highly scalable and energy efficient	Increases the complexity and control overhead
H-LEACH [24]	Chooses the CH based on the residual and minimum energy	Overcomes the node energy issues	Increases the complexity and control overhead
P-LEACH [25]	Combines PEGASIS and LEACH protocols	Energy efficiency and lifetime enhanced, traffic overload reduced	Increases the complexity and message overhead
MHT-LEACH [26]	Divides the network into internal group (send data directly to BS), and external group (establish a special routing table)	minimize distance used to send data	use only two-hop communication between external group, and internal group
IMHT-LEACH [26]	Multi-hop communication clustering protocol that consists of 4 phases	Reduced the total energy consumption	Bad Choice of CH nodes, and have an unequal number of nodes in the clusters
Enhanced LEACH [27]	The distances calculated from a node to every CH and from CH to base station and the minimum distances chosen	Minimizes the distance of transmission	Suffers from all other defects that face LEACH
ESO-LEACH [28]	Uses advanced nodes with initial energy to divide the field into smaller zones	Advanced nodes helps in extending network lifespan	Use of advanced nodes makes it expensive
IH-LEACH [29]	Combines between LEACH protocol and D-HEED [31] protocol	extends the lifespan of the network	Consumes much energy
Modified-LEACH [31]	Added some factors to alter threshold T(n) to choose the proper CH nodes	Enhances the power utilization and congestion within the network	Did not considered the privacy and security of the nodes

its geographic location and network geographic boundaries. Advantages of LBDD are that it is very simple to determine and establish, the line structure is easily accessible by the source nodes and the sink, and the overhead of these operations is low. Disadvantages of LBDD are that it still relies on broadcasts for propagating data queries along the line, and the line has to be wide enough to mitigate hotspots; therefore, especially for large networks, the flooding on the line will cause a significant increase in total energy consumption.

E. TREE-BASED CLUSTERING ROUTING PROTOCOLS

Reliable and Energy-Efficient Multi-Hop LEACH-Based Clustering Protocol for Wireless Sensor Networks Enhanced multi-hop LEACH (EM-LEACH) [36], is a tree-based routing structure, this algorithm uses new rules for CH selection and round time computing. Each node in the network randomly decides whether to become a CH for the current round. This decision is made by using two factors; a random real number factor, and the residual energy factor for selecting the suitable cluster-head. EM-LEACH proposes a variable round time, its adapts the round time according to the remaining energy in the entire network without extra overhead. EM-LEACH integrates the multi-hop communication model by using a leveling phase and a generic multi-hop routing process. Advantages of EM-LEACH it uses the variable round time to reduce overhead. Disadvantages of EM-LEACH is, the selection of CH depend only on a random number, and the residual energy. Table 2 depicts a comparison between the aforementioned clustering routing protocols.

IV. THE PROPOSED CONGESTION AWARE CLUSTERING AND ROUTING (CCR) PROTOCOL

The main objective of the proposed CCR protocol is to avoid congestion by introducing low overhead, load distribution,

TABLE 2. Comparison between clustering protocols.

Protocol	Scalability	Stability	Self-organization	Control	Hop-count	Efficiency	Homogeneous	location	Aggregation	Delay
LEACH	Low	Medium	✓	D	1	Low	✓	×	✓	V.Small
LEACH-C	Good	Medium	✓	C	1	Medium	✓	✓	✓	Small
LEACH-F	Low	High	×	C	1	High	✓	✓	✓	Small
TL-LEACH	V.Good	Medium	✓	D	2	Medium	✓	✓	–	Medium
LEACH-ET	V.Good	–	✓	D	1	Medium	✓	✓	✓	–
E-LEACH	V.Good	Medium	✓	D	1	High	×	✓	✓	Small
MB-LEACH	V.Good	–	✓	D	M	V.High	✓	✓	✓	–
TB-LEACH	–	–	✓	D	1	High	✓	×	✓	–
A-LEACH	Good	–	✓	D	1	Medium	×	×	✓	–
LEACH-H	–	–	–	H	1	Medium	✓	✓	×	–
R-LEACH	–	–	–	D	M	High	–	✓	✓	–
V-LEACH	V.Good	–	✓	D	1	V.High	✓	✓	–	–
CELL-LEACH	V.Good	–	✓	D	M	V.High	✓	✓	–	–
H-LEACH	–	–	–	D	1	V.High	✓	×	×	–
P-LEACH	–	–	–	–	M	V.High	✓	✓	✓	–
MHT-LEACH	Good	High	✓	D	M	High	✓	×	✓	Medium
IMHT-LEACH	V.Good	High	✓	D	M	V.High	✓	✓	✓	Medium
Enhanced LEACH	Low	Medium	✓	D	1	Low	✓	×	✓	Small
ESO-LEACH	Low	Medium	✓	D	1	Medium	×	×	✓	Small
IH-LEACH	Low	Medium	✓	H	1	Medium	✓	×	✓	–
Modified-LEACH	Good	High	✓	D	1	High	✓	×	✓	Medium
TEEN	Good	High	–	D	1	V.High	✓	×	×	Small
PEGASIS	Good	Low	–	–	M	High	✓	×	✓	V.Large
PANEL	Low	Low	–	D	M	High	✓	✓	✓	Moderate
LBDD	Low	Medium	✓	–	M	Medium	✓	✓	✓	Medium
EM-LEACH	Good	Medium	✓	D	M	Medium	✓	×	✓	Medium

stability, reliability, scalability and fault tolerance. Figure 6 shows the sequence of the operation of the proposed CCR protocol. The operation of the proposed CCR protocol consists of a set of rounds. Each round consists of a setup phase and a transmission phase. A setup phase is done once at the beginning of the first round, in this phase the network area is divided into levels L and sectors S to create fixed clusters. The intersection of each level and sector creates a cluster, each cluster has its PCH and SCH. The remaining rounds start with a small setup phase where the roles of the PCH node and the SCH node are rotated among cluster nodes. The transmission phase consists of two levels: intra-cluster routing, and inter-cluster routing. In the following subsections, these phases will be discussed in more details.

A. SETUP PHASE

The setup phase is executed only once in the first round where fixed clusters are organized. The created clusters remain fixed for the remaining rounds, only the rotation of the PCH and SCH roles are performed at the beginning of each round. At the end of this phase, the network area is divided into levels and sectors. The intersection of each level and sector creates a cluster. Every node must also know its cluster number, which consists of a level number L followed by sector number S denoted as C_{ls}. Each cluster must have a PCH node and optionally SCH node. The PCH and SCH nodes are selected in the setup phase depending on the distance. Other metrics,

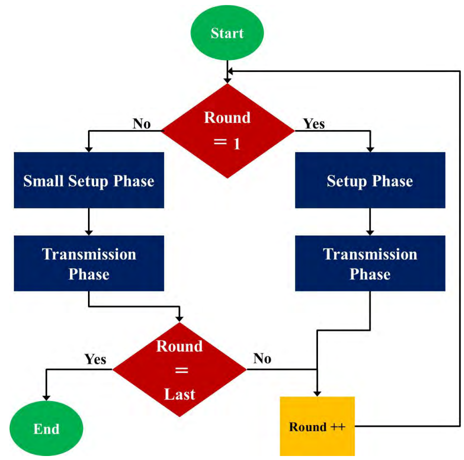


FIGURE 6. Sequence of operation of the proposed CCR protocol.

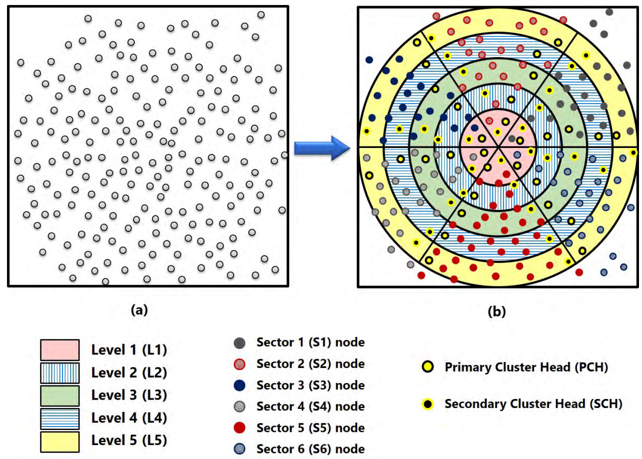


FIGURE 7. (a) 250 node distribute randomly in network area. (b) Cluster configuration.

such as energy, are not considered because, in the first round, all nodes have full energy. Figure 7 (a) shows 250 nodes, which are distributed randomly in the network area, and Figure 7 (b) shows the network area structure at the end of the setup phase, where clusters with their PCH and SCH are organized.

Algorithm 1 shows the sequence of operation of the setup phase gradually as follows: (1) Formulation of the level is done, in which each node knows its level number L_i. (2) Using the number of nodes in level L_1 to determine the number of sectors. (3) Selecting PCH_{11} and clustering other nodes in L_1 depending on their distance from PCH_{11}. (4) Equalizing the number of nodes in the clusters of level L_1. (5) Choosing SCH node in L_1 clusters. Finally, to cluster other levels: each level depends on the created clusters of the pre-level, on the equalized nodes of the clusters in this level and on the chosen SCH.

1) LEVEL FORMATION

The level number L_I indicates the hop distance between a specific node and the sink node. First, all nodes have $L = 0$

Algorithm 1 Setup Phase Algorithm

Input: N // nodes info
Output: *Cluster* // a matrix containing cluster data (PCH, SCH and other nodes

$L \leftarrow 0, D \leftarrow empty, S \leftarrow 0$
$MHM \leftarrow$ Max Hello Messages
foreach $n \in Network$ **do**
  $HM \leftarrow$ Hello Messages received from Sink
  $L = MHM - HM + 1$
  $D \leftarrow d_n$ // distance from Sink
end
foreach $n \in L1$ **do**
  Sink $\leftarrow$ "L1_Node" message
  $N_{L1} \leftarrow N_{L1} + 1$
end
d=max(D)
$N_S \leftarrow max_{NS}(d, N_{L1})$
$PCH_{11} \leftarrow node(d)$
foreach $S \in L1$ **do**
  choose PCH_{1S}
  $Cluster.nodes \leftarrow$ same no. of nodes in S
  $SCH_{1S} \leftarrow max_d(PCH_{1S})$
  $Cluster.CH \leftarrow (PCH_{1S}, SCH_{1S})$
end
foreach $L > L1$ **do**
  foreach $S \in L$ **do**
    $n_c \leftarrow min_d(C_{L-1,S}, C_{L,S-1})$
    arrange n_c
    $PCH_{LS} \leftarrow min(n_c)$
    broadcast PCH_{LS}
    $Cluster.nodes \leftarrow$ same no. of nodes in S
    $SCH_{LS} \leftarrow max_d(PCH_{LS})$
    $Cluster.CH \leftarrow (PCH_{LS}, SCH_{LS})$
  end
end
return *Cluster*

and $S = 0$. The sink node initiates the determination of the level number of each node. It broadcasts a "HELLO" message many times, each time with a different radius until it reaches the end of the network. According to the number of received messages, each node specifies its level. Nodes in the last level receive one message, while nodes in the first level receive all the messages.

2) CLUSTERING FIRST-LEVEL NODES

In the CCR protocol clusters are fixed for all rounds, and all levels have an equal number of sectors, so the number of sectors in each level must be calculated.

a: CALCULATE THE NUMBER OF SECTORS (NS)

Each node in L_1 transmits back to the sink an "L_1-NODE" message. The sink node determines the number of nodes in

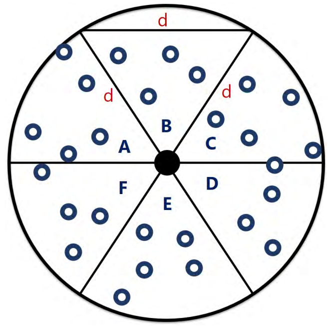


FIGURE 8. L_1 **Circle and triangle area.**

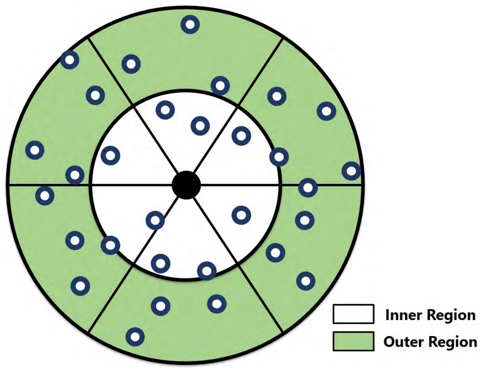


FIGURE 9. **Outer and inner region.**

L_1 denoted as N_{L1} after receiving this message. It uses N_{L1} to determine the number of sectors (NS) for L_1, assuming that all levels have the same number of sectors. To calculate NS, the distance (d) between the sink node and the farthest node in L_1 is determined. The sink determines the distances to the nodes using the received signal strength of "L_1-NODE" message. The distance d is used to calculate the circle area of level L_1. An equilateral triangle with side length equals d is calculated, as shown in Figure 8. The maximum number of sectors is determined by dividing the circle area by the triangle area as Equation (1).

$$Max(NS) = \frac{\pi d^2}{((\sqrt{3}d^2)/4)} \tag{1}$$

From Equation (1), the maximum NS equals 7.2, therefore the maximum number of sectors in each level will not exceed 7 sectors.

b: DIVIDING L_1 TO INNER AND OUTER REGIONS

To cluster L_1, the level is divided into two regions; inner and outer regions as shown in Figure 9. PCH nodes are located at the outer region. Nodes in the inner region are MNs unless there is not PCH at the outer region.

c: CHOOSING PCH_{11} AND CLUSTERING OTHER NODES

The sink assigns the farthest node in the outer region of L_1 as the PCH of C_{11} denoted as PCH_{11}. It sends a "HEAD" message to PCH_{11}, including level number, sector number

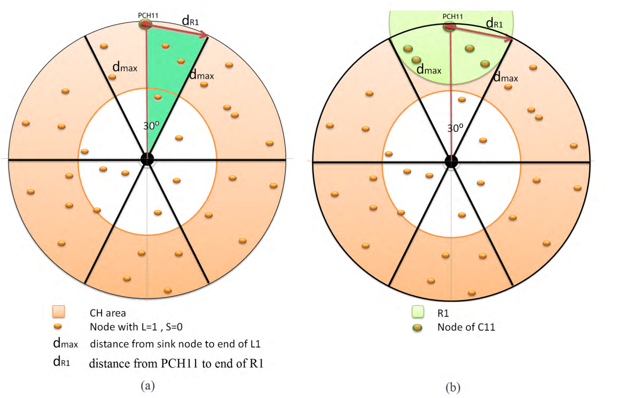


FIGURE 10. (a) Calculate d_{R_1}. (b) Join member nodes to C_{11}.

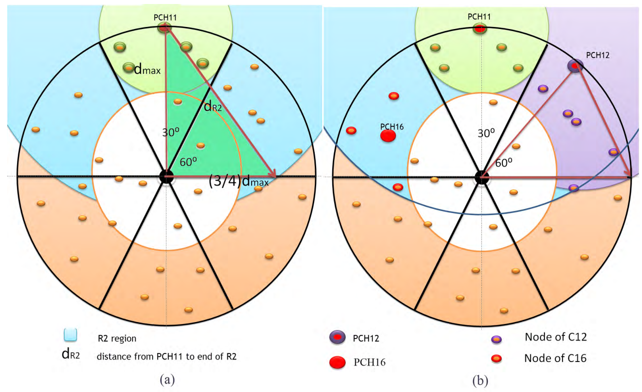


FIGURE 11. (a) Calculate d_{R_2}. (b) Join member nodes to C_{12} and C_{16}.

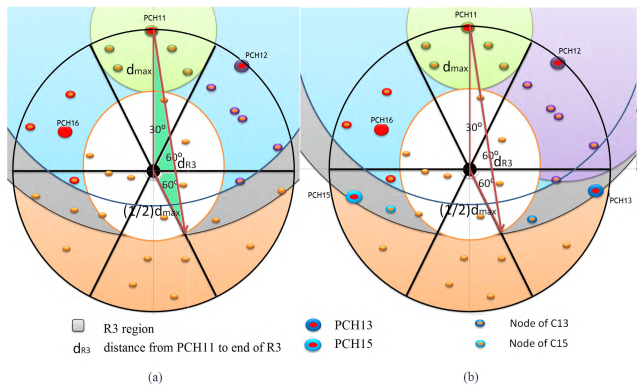


FIGURE 12. (a) Calculate d_{R_3}. (b) Join member nodes to C_{13} and C_{15}.

and the number of nodes in the cluster. PCH_{11} broadcasts a message to all nodes in a distance equals double the distance between the sink and PCH_{11} ($2 \times d_{max}$). Each node in L_1, on receiving the message, determines its sector based on the distance to PCH_{11}.

The proposed CCR Protocol divides the network area into six sectors. To create six equal sectors, mathematical equations are used to divide the network area into five regions, each region R_i depends on the distance from that of the PCH_{11}. These five regions are used to organize the six equal clusters in L_1 as follows.

REGION1 (R_1)

R_1 is used to find the area of C_{11}. Figure 10 (a) shows a green isosceles triangle with edge length equal to the d_{max} length, the vertex angle of 30 degrees and a base angle of 75 degrees. In this triangle, the length of the d_{R_1} is calculated using Equation (2), which is the radius of R_1 circle. As shown in Figure 10 (b), all nodes of L_1 join C_{11} if they are located in the circle with radius d_{R_1}, and they are not assigned to any other cluster yet.

$$d_{R_1} = 2 \times d_{max} \times cos((180-30)/2) = 2 \times d_{max} \times cos(75) \quad (2)$$

REGION2 (R_2)

R_2 is used to find the area of C_{12}, and C_{16}. Figure 11 (a) shows the green triangle which has one edge that is equal to the d_{max} whereas the other edge is equal to $(3/4)d_{max}$ and the angle between them is 90 degrees. In this triangle, the length of d_{R_2} is calculated using Equation (3), which is the radius of R_2 circle. As shown in Figure 11 (b), all the nodes of L_1 join C_{12} or C_{16} if they are located in the circle with radius d_{R_2} and yet are not joined with any other cluster. First, PCH_{11} chooses the closest node to it from R_2 to be the PCH_{12}. Then the PCH_{12} chooses the closer nodes to it from R_2 to be the members of C_{12}. Other nodes in R_2 become members of C_{16} and any node of them is chosen as the PCH_{16}.

$$d_{R_2}{}^2 = d_{max}{}^2 + (\frac{3}{4}d_{max})^2 - \frac{3}{2}d_{max}{}^2 \times \cos(90) \quad (3)$$

REGION3 (R_3)

R_3 is used to find the area of C_{13}, and C_{15}. Figure 12 (a), shows the green triangle, which has one edge that is equal to the d_{max} whereas the other edge is equal to (1/2) d_{max} and the angle between them is 150 degrees. In this triangle, the length of d_{R_3} is calculated using Equation (4), which is the radius of the R_3 circle. As shown in Figure 12 (b), all the nodes of L_1 join C_{13} or C_{15} if they are located in a circle with radius d_{R_3}, and yet are not joined with any other cluster. First, PCH_{12} chooses the closest node to it from R_3 to be the PCH_{13}. If there are no nodes in C_{12}, the PCH_{11} takes this role of the chosen PCH_{13}. Then PCH_{13} chooses the closer nodes to it from the R_3 to be the members of the C_{13}. Other nodes in R_3 become members of the C_{15} and any node of them is chosen as the PCH_{15}.

$$d_{R_3}{}^2 = d_{max}{}^2 + (\frac{1}{2}d_{max})^2 - d_{max}{}^2 \times \cos(150) \quad (4)$$

REGION4 (R_4)

R_4 is used to find the area of C_{13}, C_{14}, and C_{15}. Figure 13 (a) shows the green isosceles triangle with edge length equal to the d_{max} length, the vertex angle of 150 degrees and a base angle of 15 degrees. In this triangle, the length of d_{R_4} is calculated using Equation (5), which is the radius of the R_4 circle. As shown in Figure 13 (b), all nodes of L_1 join C_{13},

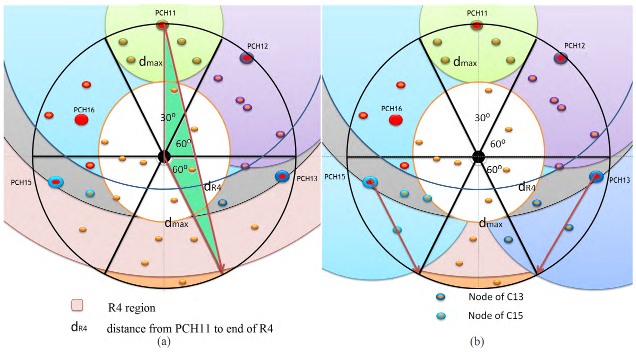


FIGURE 13. (a) Calculate d_{R_4}. (b) Join member nodes to C_{13}, C_{14}, and C_{15}.

C_{14}, or C_{15} if they are located in a circle with radius d_{R_4}, and yet are not joined with any other cluster. First, the PCH_{13} chooses the closer nodes to it from R_4 to be the members of the C_{13}. Then, PCH_{15} chooses closer nodes to it from the R_4 to be the members of the C_{15}. Other nodes in R_4 become members of the C_{14} and any node of them is chosen as the PCH_{14} if there are no nodes in R_5.

$$d_{R_4} = 2 \times d_{max} \times cos(15)) \tag{5}$$

REGION5 (R_5)

R_5 is used to find the area of the C_{14}. Figure 14 shows the green triangle, which has two equal edges that are equal to the d_{max} and the angle between them is 180 degrees. In this triangle, the length of d_{R_5} is calculated using Equation (6), which is the radius of the R_5 circle. As shown in Figure 14, all nodes of L_1 join C_{14} if they are located in a circle with radius d_{R_5}, and yet are not joined with any other cluster.

$$d_{R_5} = 2 \times d_{max} \tag{6}$$

d: ADD NODES IN THE INNER REGION TO CLUSTERS

If all clusters have a PCH node, each node in the inner region sends a request to join to the closest PCH. Otherwise, if there is one cluster that does not have a PCH, the inner region will be divided again into two regions depending on PCH_{11}.

Any node in the inner region, with a distance to PCH_{11} less than the distance between PCH_{11} and a sink node, can be determined as the inner region 1. However, other nodes can be determined as the inner region 2. Figure 15 shows the inner region, which is divided into two almost equal regions. If the PCH node is not found in the clusters (C_{16}, C_{11}, or C_{12}), this missed PCH is taken from the inner region 1. If the PCH node is also not found in clusters (C_{13}, C_{14}, or C_{15}), this missed PCH is taken from the inner region 2. Afterward, the remaining nodes in the inner regions, request to join the cluster of the closest PCH.

e: EQUALIZING CLUSTERS

Each cluster should contain an equal number of nodes. Therefore, each cluster calculates the optimal number of nodes,

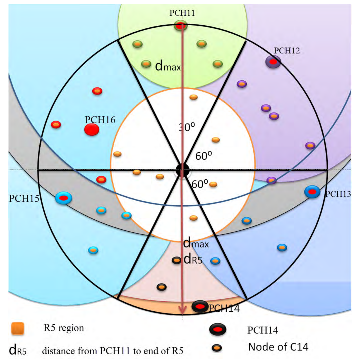


FIGURE 14. Calculate d_{R_5} and join member nodes to C_{14}.

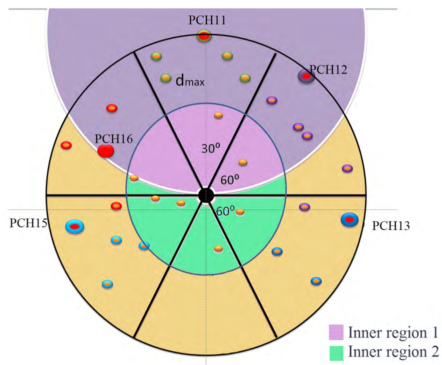


FIGURE 15. Inner region 1 and inner region 2.

to make all the clusters, approximately, have the same number of nodes. Equation (7) is used to equalize clusters.

$$NC_{LS} = \lceil \frac{N_L}{N_S} \rceil \tag{7}$$

where:

- NC_{LS}: Number of nodes of the cluster in level (L) and sector (S).
- N_L: Number of nodes in level (L).
- N_S: Total number of sectors.

Each PCH sends the excess nodes to neighbor clusters. Afterward, all clusters in level L_1 will have an equal number of nodes.

f: CHOOSE SCH

In each cluster, the farthest node from PCH node is chosen as an SCH, as shown in Figure 16, which is used to help PCH in its role in the transmission of data.

3) CLUSTERING LEVEL(X) NODES

All nodes in the network know their level according to the number of "HELLO" messages received, previously. Each PCH in level L_1 specifies nodes of level L_2 in the range of d_2 according to Equation (8).

$$d_n = d_{max}(L_n) - d_{max}(L_{n-1}) \tag{8}$$

where:

- n: Number of the next level.

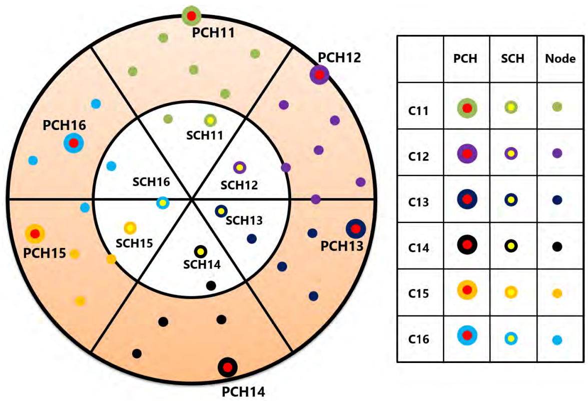


FIGURE 16. Choose SCH in each cluster in L_1.

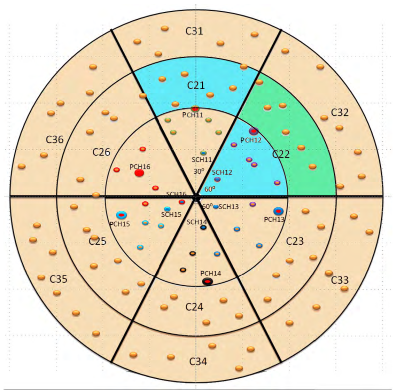


FIGURE 17. New cluster C22 chosen to be closer to clusters C12 and C21.

- d_n: the range of level L_n.
- $d_{max}(L_n)$: the max radius of level L_n.

CHs of next level are located in this region, where PCH is chosen as the closest node to PCH of pre-cluster. PCH of the new cluster broadcasts itself on the network. Figure 17 shows that the nodes of each cluster are chosen to be close to two clusters (the pre-level cluster in the same sector and the preceding sector cluster in the same level). Each cluster equalizes the number of nodes, then chooses SCH as explained before. This step is repeated until all nodes in a network join a specific cluster with the specific level number and specific sector number.

4) CLOSING THE SETUP PHASE

When all nodes of the last level know their clusters, this means that the "SETUP" phase is done. During this phase, a Carrier Sense Multiple Access (CSMA) MAC protocol is being used. CSMA allows nodes to sense the channel, if empty, before sending any message. At the end of this phase, the network area is divided into levels and sectors to form clusters, and all nodes know their cluster number.

5) THE COMPLEXITY OF SETUP PHASE

In this section, the complexity of implementing and running the Setup phase is studied in both terms of communication costs and time costs. Communication costs can be described using the number of control messages as shown in Equation (9).

$$CM = 2 \times NS + nodes(L_1) + \left\{ \sum_{i=2}^{i=NL} 2 \times nodes(L_i) \right\} + NS \times L_{max} \tag{9}$$

where:

- CM: Number of Control Messages.
- NS: Number of sectors.
- NL: Number of levels.
- $nodes(L_i)$: Number of nodes in level L_i.

The time cost is calculated by summing up the time of sending messages between nodes, headers, and the sink, as shown in Equation (10).

$$T = NL \times (NS - 1) \times t_c \tag{10}$$

where:

- T: Total time of Setup phase.
- t_c: number of nodes in each cluster * time to send messages.

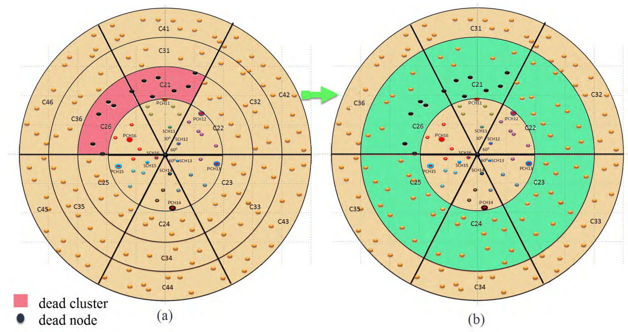


FIGURE 18. Merge of two levels.

B. THE SMALL SETUP PHASE

This phase is executed at the beginning of each round other than the first round. It has four main steps, explained in the next subsections. Algorithm 2 shows the sequence of the "SMALL SETUP" phase.

1) REMOVING THE DEAD NODES

When each node sends its "NODE-CONDITION" message to its PCH, if any node has less power than the minimum energy required for transmitting and receiving signals, this node is denoted as a dead node and is removed from the cluster.

2) SOLVE THE PROBLEM OF THE DEAD CLUSTERS

If any level except the last one has a dead cluster, it merges its clusters with clusters in the next level. Figure 18 shows the merging of clusters of two levels when one of them has dead clusters.

Algorithm 2 Small Setup Phase Algorithm

```
Input: d_n, D_C, N_n // dead nodes, dead clusters, New nodes
foreach d_c ∈ D_C do
    if L(d_c) < Lastlevel then
        MergeLevel(L(d_c), L(d_c + 1))
        n ← 1
    end
end
foreach l ∈ L do
    foreach c ∈ l do
        foreach n ∈ c do
            PCH ← "Node_condition" message
            if PCH return ACK then
                PCH find new PCH,SCH
            else
                SCH ← "Node_condition" message
                SCH return ACK
                SCH find new PCH,SCH
            end
            Send "PCH_Head" to new PCH
            Send "SCH_Head" to new SCH
            Broadcast PCH, SCH
        end
    end
    foreach c ∈ l do
        if N_n(c) ≠ Empty then
            c ← N_n(c)
        end
    end
end
```

3) FINDING NEW PCH AND SCH NODES

PCH node is chosen depending on the Remaining Energy (E_R), the amount of free space of the storage unit in each node (Free Queue Length) (FQL), the amount of data that each node wants to send (S_D), the distance from each node to the PCH node of the cluster in [the previous level and the same sector] (d_P) and the distance from each node to the SCH node of the cluster in [the previous level and the same sector] (d_S). Transmission in each cluster in the "SMALL SETUP" phase is done using CSMA and CDMA in MAC protocol. CDMA prevents collisions of data sent between two sides (clusters, sectors,..) through using a specific code to send data with.

Choosing new CH nodes is done sequentially level by level in six consecutive steps. In the first step, when the time of the new round starts, the sink node broadcasts the Start Round Number "START-RN" message. When the nodes in L_1 receive this message, each node in each cluster sends a "NODE-CONDITION" message to the old PCH of its cluster, which acknowledges this message. If any node does not receive the acknowledge message from the old PCH, it sends its "NODE-CONDITION" message to the old SCH. The structure of a "NODE-CONDITION" message is shown in Figure 19.

R_n	L_n	S_n	N_{id}	E_R
FQL	S_D	d_P	d_S	

FIGURE 19. The structure of NODE-CONDITION message.

The Remaining Energy (E_R) is used to find the best CH nodes with a large amount of energy. The amount of data that each node wants to send (S_D) is used to find the CH nodes with the biggest amount of data to send. If a node with big data is chosen as the PCH; the overhead of sending its data will disappear, and the total amount of data, which will be sent in intra-cluster communication, will be minimized. Free queue length (FQL) is used to find the best PCH nodes with FQL, which can contain half the amount of the sent data, data are distributed between the PCH node and the SCH node. (d_P and d_S) factors are used to save the power of communication in the inter-cluster routing by minimizing the distance between the CH nodes.

In the second step, the old PCH or the old SCH uses E_R, FQL, S_D, d_P, and d_S of each node in the cluster to find the best two nodes to be the new PCH, and SCH using Equations (11) and (12). However, in the level (1), d_P and d_S are the factors that contain the distance from each node to the old CH and SCH in the same cluster.

$$PCH_{new} = n \in N \,|\, \max_{E_R}(n) \wedge \max_{S_D}(n) \wedge \min_{d_P}(n) \quad (11)$$

$$SCH_{new} = n \in N \,|\, \max_{E_R}(n) \wedge \max_{S_D}(n) \wedge \min_{d_S}(n) \quad (12)$$

where:

- N the set of nodes in the network.
- $\max_{E_R}(n)$ node with the maximum remaining energy.
- $\max_{S_D}(n)$ node with the maximum data needed to be sent.
- $\min_{d_P}(n)$ node with the minimum distance to PCH.
- $\min_{d_S}(n)$ node with the minimum distance to SCH.

In the third step, the Old PCH sends a "P-HEAD" message to the new PCH node and an "S-HEAD" message to the new SCH node. It also sends the amount of data that each node wants to send (S_D). In the fourth step, the New PCH broadcast itself by using (PCH of cluster number for round numbers) a "P-HCLS-RN" message, this message is sent using CSMA and not CDMA to enable any node to receive it. In the fifth step, the new SCH broadcasts itself by using (SCH of cluster number for round numbers) an "S-HCLS-RN" message, this message is sent using CSMA and not CDMA to enable any node to receive it. In the sixth step, when each node in level L_2 receives "P-HCLS-RN" and "S-HCLS-RN" messages of the cluster in [the previous level and the same sector], it computes the distance to the preceding PCH and SCH. Put this distance in (d_P) and (d_S) factors in the "NODE-CONDITION" message which also contains E_R, FQL, and S_D. Then, it sends this "NODE-CONDITION" message to the old PCH in its cluster. Repeat all steps until all the clusters have new PCH and SCH.

4) ADDING NEW NODES

Any new node, which is not assigned to a level or a sector number and receives a "P-HCLS-RN" messages, chooses the nearest CH node and sends a "CONNECT" message. This CH node must send an "ACK" message, then this new node changes the level and the sector number of it by the level and the sector number of the chosen CH node and takes its CDMA to communicate within the next round.

C. INTRA-CLUSTER ROUTING

In intra-cluster routing, all ordinary nodes of each cluster send their data to their PCH node using single-hop communication. Transmission in this phase is done using TDMA and CDMA. TDMA divides time between nodes. Each node sends data in its specific time slots. First, the PCH node must determine the destination node (PCH or SCH) for each node in the cluster. In addition to the first time slot for transmitting each node in the cluster, and the number of slots given to each node to send their data. Algorithm 3 shows the sequence of the intra-routing phase.

1) DETERMINING DESTINATION NODE

The destination node of PCH node is itself, and the destination node of SCH is itself. Other nodes, specify two distances; the first is the distance to PCH + distance between PCH and Sink. The second is the distance to SCH + distance between SCH and Sink. Each node sends its info message to the PCH node. This message contains the remaining energy of a node, the two distances that are calculated and the amount of data that needs to be sent. The PCH node knows the free size of its storage unit and the storage unit of SCH, as well. For each node, the PCH node uses two calculated distances to find a closer CH node, then tests the size of the CH, which has the closer distance, and checks if the closer CH can take the data from this node or not. If this closer CH can take all the data from this node, then the destination CH of this node is the closer CH. If the size of this closer CH is smaller than the size of the data that the node wants to send, it only takes the size that is necessary to complete its storage unit and makes the destination CH node of this part of the data closer CH. The remaining data checks if it can find a place in the far CH. If it finds a place, the destination CH node for the remaining data becomes the farthest CH. If the closer CH does not have any free size of its storage unit, the CH checks the farthest CH node's free size.

2) DETERMINING THE NUMBER OF SLOTS GIVEN TO EACH NODE

Other protocols give an equal number of slots to each node to send its data to the CH node regardless of the amount of data sent by each node. In the proposed CCR protocol, the slots given to each node depend on the amount of data sent by each node. The PCH must determine two factors for each node. The first factor is the (First-Slot-Time) (FST) which means the time when each node sends its first slot. The second factor is the Number of Slots (NoS) given to each node. This number depends on the amount of data that each node wants to send.

Algorithm 3 Intra-Routing Algorithm

Input: C `// Specific cluster`
Output: P_{data}, S_{data} `// PCH-data, SCH-data`

foreach $n \in C$ **do**
  $d_p \leftarrow dist(n, PCH) + dist(PCH, Sink)$
  $d_s \leftarrow dist(n, SCH) + dist(SCH, Sink)$
  $n_d \leftarrow$ node data size
  send d_p, d_s, n_d to PCH
  $f_p \leftarrow freeSize(PCH)$
  $f_s \leftarrow freeSize(SCH)$
  if $d_p < d_s$ **then**
    fill f_p first, then f_s
  else
    fill f_s first, then f_p
  end
end
$P_{data} \leftarrow PCH.aggregate()$
$S_{data} \leftarrow SCH.aggregate()$
$f_p \leftarrow freeSize(PCH)$
$f_s \leftarrow freeSize(SCH)$
foreach $n \in C$ **do**
  $n_d \leftarrow$ node data size
  if $n_d > 0$ **then**
    fill free(f_p, f_s)
  end
  $n_d \leftarrow$ node data size
  if $n_d > 0$ **then**
    drop n_d
  end
end
return P_{data}, S_{data}

3) SENDING DATA

In each cluster, the PCH node broadcasts a "TDMA table" message, which contains a destination node (PCH or SCH) for each node in the cluster, a first slot time for each node in the cluster and the number of slots given to each node in the cluster. Table 3 shows the structure of the "TDMA table" message.

Each node sends its data to the PCH nodes in its slots and goes into the sleep mode for the remaining slots. The PCH node is awake almost all the time in an intra-cluster routing.

4) AGGREGATING DATA

When all nodes send their data to their CH node, the CH nodes (primary and secondary) aggregate this data to send them in the inter-cluster routing.

TABLE 3. TDMA table message.

NODE_ID	Destination	FST	NoS
Id of node	PCH or SCH	Time of first slot	Number of slots given to this node

5) TRYING TO SEND REMAINING DATA

Free size of data in the PCH and SCH nodes begins to appear after aggregation. Thus, if there are data that are not sent to the CH nodes since they complete their storage unit before aggregation, these data try to find places in the PCH or SCH after aggregation. This is done to minimize the size of the dropped data.

D. INTER-CLUSTER ROUTING

In this phase, each CH node tries to send its data to the sink node. Each sector sends its data to the same sector in pre-level until the data reach the sink node, so each sector at all levels uses the same CDMA to send the data. The data are also aggregated again in each CH from its way to the sink node. Algorithm 4 shows the sequence of inter-routing.

At the beginning of this phase, only the PCH node and the SCH node are awake. The CH-NODES for each cluster broadcast an "RN-CH" (round number cluster head) message using CSMA and CDMA for the same cluster at all levels. This message contains a round number, a level number, a sector number, its ID, the amount of data it needs to be sent and the amount of free space storage unit in each node (free queue length) (FQL). Transmission is divided into several periods, according to the size of data that each level needs to send, and the number of PCH and SCH node in level (1). The transmission in each period is done when all the PCH and SCH nodes of the level (1) are full of data.

For example: Using a network that consists of 7 levels and 6 sectors where all the sectors in level (1) have PCH and SCH. The Data, which are sent in every period, must not exceed the size of all the PCH and SCH of level (1) nodes, to avoid excess data. The end level of each period is determined using the condition of [Size of data sent < no of PCH and SCH of the level (1) * max storage unit].

- Period 1: The data of (level (1) and level (2)) < number of PCH and SCH of the level (1) * max storage unit Therefore, level (2) sends the data to level (1), level (1) aggregates the data and sends it to the sink node.
- Period 2: The data of (level (3, 4, 5)) < the number of PCH and SCH of the level (1) * max storage unit Thus, level (5) sends the data to level (4), level (4) aggregates the data and sends it to the pre-level and so on until it reaches the sink node.
- Period 3: The data of (level (6,7)) < the number of PCH and SCH of the level (1) * max storage unit. Therefore, level (7) sends the data to level (6), level (6) aggregates the data and sends it to the pre-level and so on until it reaches the sink node.

Algorithm 4 Inter-Routing Algorithm

Input: P_{data}, S_{data} `// PCH-data, SCH-data`
Output: $total_d$ `// Total data`

$max_s \leftarrow dsize_{L1}(PCH) + dsize_{L1}(SCH)$
for $L \leftarrow 1$ **to** MAX **do**
  $dsize_{L_{max}} = dsize_L(PCH) + dsize_L(SCH)$
end
while $dsize_{L_{max}} \neq 0$ **do**
  $le \leftarrow 1$
  $total_d \leftarrow data(L \leftarrow le)$
  for $L \leftarrow 1$ **to** MAX **do**
    if $total_d \geq max_s$ **then**
      send $total_d$ to previous levels to Sink
      for $RL \leftarrow L$ **to** 1 **do**
        $dsize_{L_{max}} = dsize_{L_{max}} - (dsize_{RL}(PCH) + dsize_{RL}(SCH))$
      end
      break
    else
      $le \leftarrow le + 1$
      $total_d \leftarrow total_d + data(L \leftarrow le)$
    end
  end
end
return $total_d$

The number of periods in each round is not fixed because it depends on (1) the size of the data of each round, (2) the number of nodes in each level, (3) the number of PCH in level L_1.

V. EXPERIMENTAL RESULTS: DISCUSSION AND ANALYSIS.

MATLAB simulations are used where the sensor nodes are randomly distributed in the plane region. In this case study, we compare our protocol with the LEACH, MHT-LEACH and IMHT-LEACH protocols in different scenarios; each scenario has a different network area and a different number of nodes. The experiments show that the CCR protocol outperforms the other protocols. The parameters of the simulation are listed in Table 4.

A. CCR VS. LEACH, MHT-LEACH, IMHT-LEACH

In the first comparison between LEACH, MHT-LEACH and IMHT-LEACH protocols, the sensor nodes are randomly distributed in the plane region which contains 200 nodes distributed randomly in the network area with coordination (250×250). The network area is divided into clusters containing an equal number of nodes.

The first parameter of the comparison is the number of nodes alive. The number of nodes alive shows the degree of stability of each protocol, and the length of its durability as shown in Figure 20. Clearly, the proposed protocol achieved the highest number of rounds with alive nodes. This means that the proposed protocol prolonged the lifetime of the

TABLE 4. The simulation parameters.

Parameter	Value	Parameter	Value
Initial Energy (Eo)	0.5 nJ/bit	Data aggregation	5 J/bit/signal
Transmits Energy	50 nJ/bit	Free space amplifier	10 pJ/bit/m2
Receive Energy	50 nJ/bit	Multipath amplifier	0.0013 pJ/bit/m4
P	0.1	Message size	Range (0:6400) bits

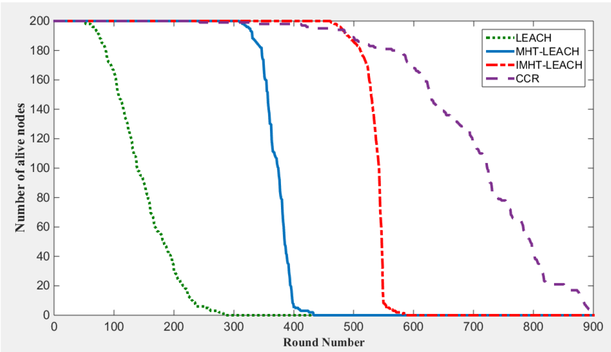


FIGURE 20. The number of alive nodes in the LEACH, MHT-LEACH, IMHT-LEACH, and CCR protocols.

network, and this is occurred due to multi-levels transmission, and the good choice of CHs.

The second group of parameters used in the comparison are the first, half and final nodes die. Figure 21 shows that the proposed protocol has a longer lifetime than other protocols. The third parameter used in the comparison is the total number of packets received by all CHs in the rounds. Figure 22 shows that the result of the highest total number of packets is accomplished by the proposed protocol. This is because the network is divided into a number of levels and sectors in the proposed protocol, choosing of best CH, and also intra, and inter routing techniques.

The fourth parameter used in the comparison is the total number of packets delivered to the BS with rounds. Figure 23 shows that the proposed protocol achieves the highest throughput compared with other protocols. The previous result is achieved due to the intra, and inter routing techniques of the proposed protocol.

B. CCR VS. LEACH

The experiments are carried out in different scenarios, each scenario has its own area and number of nodes. Table 5 shows the different scenarios used in this case study. The experiments show that the CCR protocol outperforms the other protocols. The wider the area, the better performance achieved by the CCR compared to the LEACH.

The main objective of carrying out these experiments is to prove the superiority of the proposed CCR protocol over the LEACH protocol as follows:

- The LEACH protocol creates clusters with one hop from the sink node, while the CCR protocol creates clusters with levels, which is suitable for large area networks.

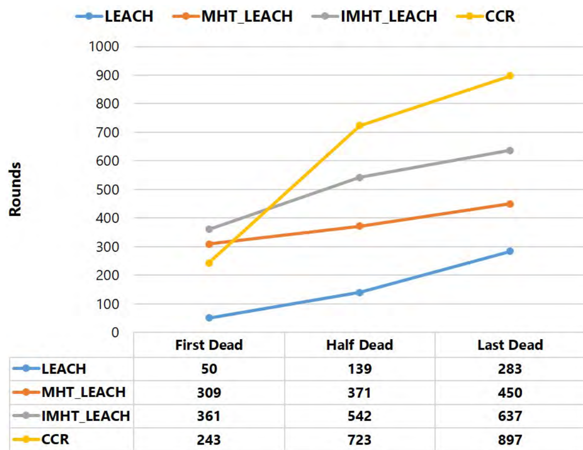


FIGURE 21. First, half, and last dead nodes in the LEACH, MHT-LEACH, IMHT-LEACH, and CCR protocols.

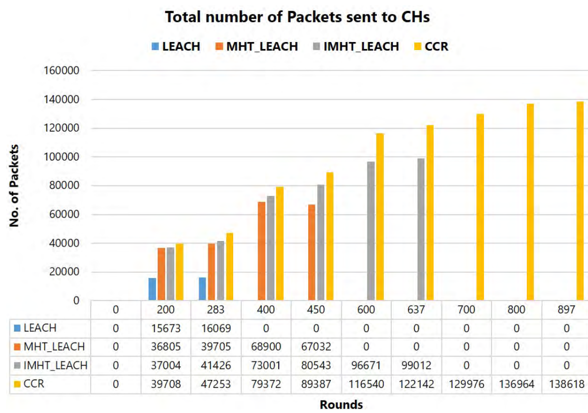


FIGURE 22. Total number of packets sent to CHs in the LEACH, MHT-LEACH, IMHT-LEACH, and CCR.

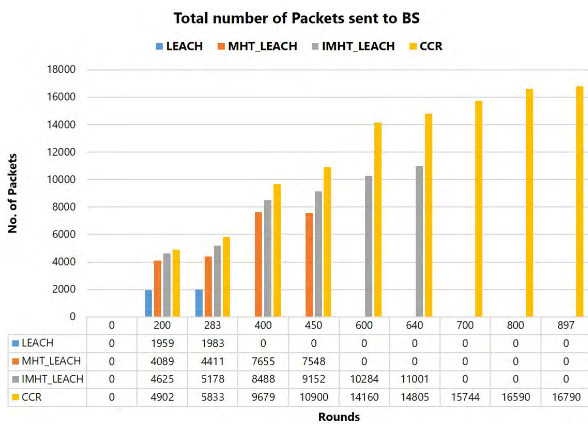


FIGURE 23. Total number of packets sent to BS in the LEACH, MHT-LEACH, IMHT-LEACH, and CCR.

- The LEACH protocol consumes much energy as it runs the setup phase, completely, at the beginning of each round. The CCR protocol runs the setup phase only once in the first round. It uses fixed clusters and changes the

TABLE 5. The simulation scenarios.

Scenario	Area	Nodes	Rounds	Nodes added	
				number	at round
1	100*100	100	1000	-	-
2	100*100	100	1000	100	500
3	200*200	150	1000	-	-
4	200*200	150	1000	100	400
5	400*400	200	500	-	-
6	400*400	200	500	100	100

role of PCH and SCH between the nodes of the cluster at the beginning of each round.

- The LEACH protocol uses one CH node, while the CCR protocol uses SCH to help PCH in data transmission. In addition, if PCH faces any problem, SCH takes its role.
- Unlike the LEACH protocol, the CCR protocol uses more than one metric in specifying PCH. These metrics are the remaining energy, the distance to pre-level PCH and SCH, the free queue length of the storage unit, and the data each node needs to send.
- The CCR protocol adds new nodes without performing a complete setup phase in each round, as in the LEACH protocol.
- In the CCR protocol, a level with dead clusters merges its nodes with the next level.
- In the intra-routing phase of the LEACH protocol, there is no action done to avoid the congestion and the overflow of the data taking place, but in the CCR protocol, there is a fault tolerance to avoid the congestion and the overflow of the data.
- In the LEACH, CH nodes send their data directly to the sink, which consumes a lot of energy, especially in large networks. In the CCR, data are sent to the sink level by level.

1) SCENARIO 1

Figure 24(a) shows the total energy of the nodes in both protocols. The total energy in the LEACH protocol degrades with a higher rate until a round where the protocol dies. The CCR protocol remains alive with approximate energy of 7J at round 1000. Figure 24(b) shows the number of flows in both protocols. The LEACH protocol suffers from a lot of overflows. These overflows cause the collision of data and re-transmission of the missed data. The CCR does not suffer from overflows due to fault tolerance features.

Figure 24(c) shows the number of packets sent to BS every round in both protocols. The total amount of data sent in the CCR protocol is larger than the LEACH protocol. The CCR protocol keeps sending data, while the LEACH protocol stops sending data as it dies at round 650. Figure 24(d) shows the number of packets dropped every round in

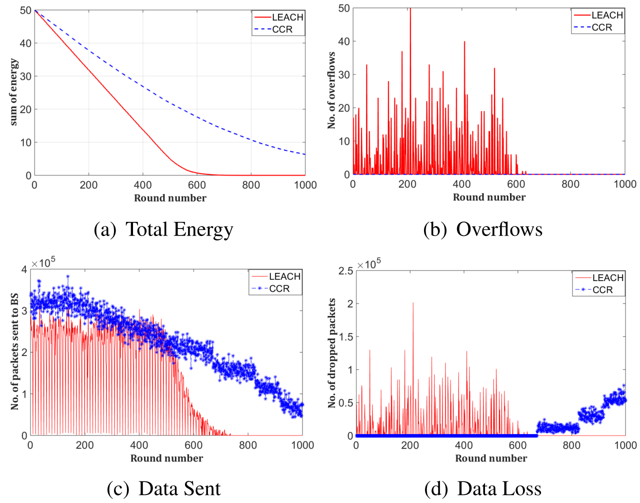


FIGURE 24. Results of the CCR protocol vs. the LEACH protocol in scenario 1. (a) Total energy. (b) Overflows. (c) Data sent. (d) Data loss.

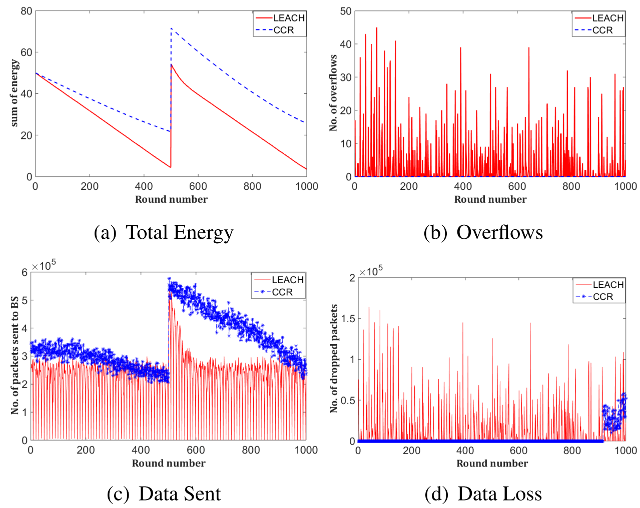


FIGURE 25. Results of the CCR protocol vs. the LEACH protocol in scenario 2. (a) Total energy. (b) Overflows. (c) Data sent. (d) Data loss.

both protocols. There is a large amount of dropped data in the LEACH. On the other hand, the CCR protocol keeps zero amount of dropped data until approximately round 680, where a small amount of data starts to drop.

2) SCENARIO 2

The total energy of the nodes in the CCR protocol is more than in the LEACH protocol. Adding the new 100 nodes at round 500 increases the total energy in both protocols as shown in 25(a). This increase happens due to the energy of the newly added nodes. Figure 25(b) shows that CCR adapts with added nodes with zero-level of overflows. On the contrary, the LEACH protocol suffers from overflows.

Adding new nodes increases the number of sent packets to BS in the CCR as shown in 25(c). It is also increased in the LEACH in the round of addition, then it decreases again due

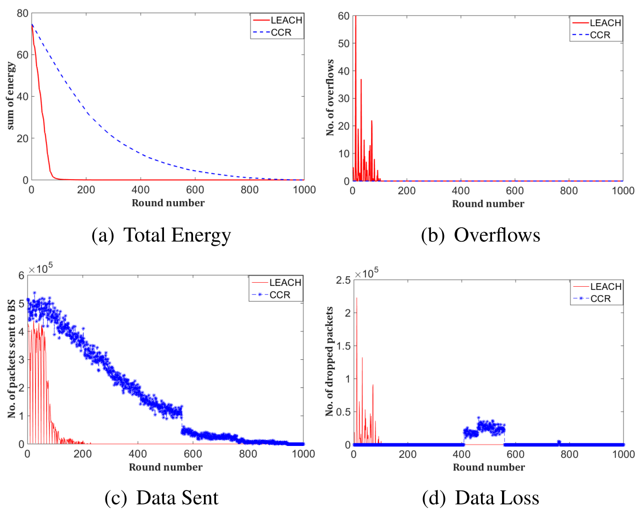


FIGURE 26. Results of the CCR protocol vs. the LEACH protocol in scenario 3. (a) Total energy. (b) Overflows. (c) Data sent. (d) Data loss.

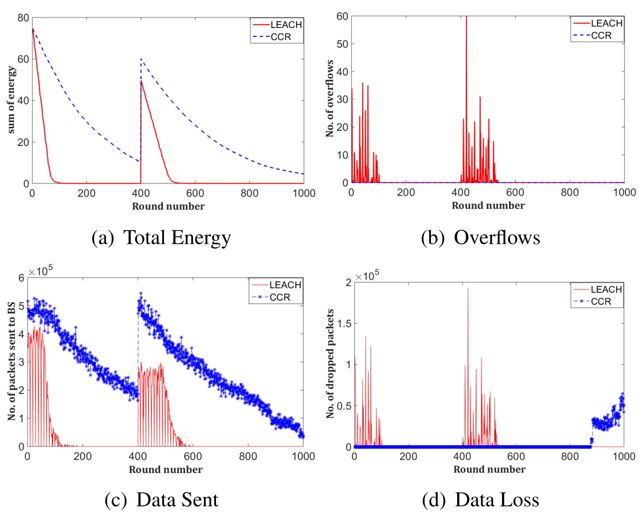


FIGURE 27. Results of the CCR protocol vs. the LEACH protocol in scenario 4. (a) Total energy. (b) Overflows. (c) Data sent. (d) Data loss.

to the number of dead nodes. Figure 25(d) shows that the CCR keeps zero amount of dropped packets until approximately round 920, where a small amount of data is dropped.

3) SCENARIO 3

Increasing the network area affects the lifetime of protocols as shown in Figure 26(a). As the area increases, the total energy of nodes in the LEACH decreases until it dies before round 100. The CCR remains alive till approximately round 950. Figure 26(b) shows that the LEACH suffers from a lot of overflows until round 100, where it dies and no messages are transmitted.

The total amount of data sent in the CCR is larger than the LEACH as shown in Figure 26(c). The LEACH stops sending data as it dies. The CCR keeps sending until round 950. Figure 26(d) shows the amount of data dropped

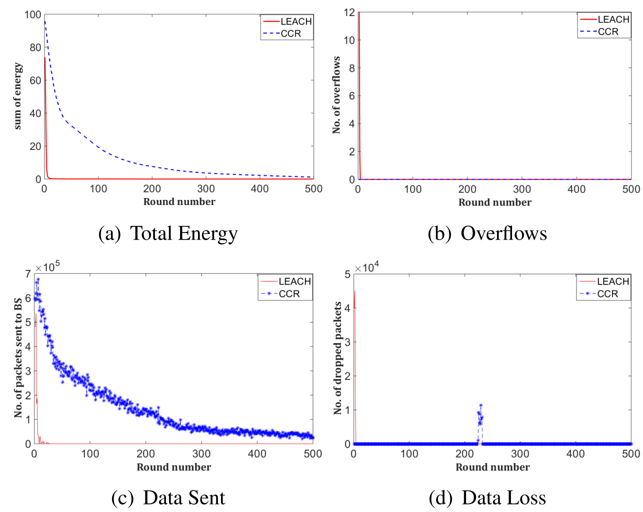


FIGURE 28. Results of the CCR protocol vs. the LEACH protocol in scenario 5. (a) Total energy. (b) Overflows. (c) Data sent. (d) Data loss.

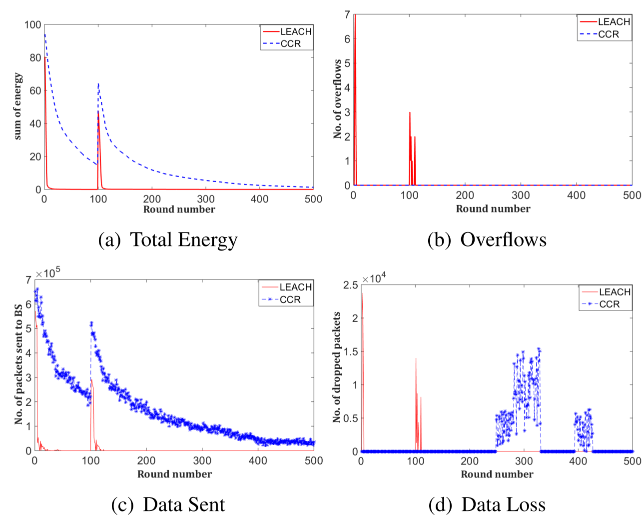


FIGURE 29. Results of the CCR protocol vs. the LEACH protocol in scenario 6. (a) Total energy. (b) Overflows. (c) Data sent. (d) Data loss.

every round. It also shows that the LEACH drops a lot of data while running, while the CCR nearly doesn't drop any data.

4) SCENARIO 4

The total energy of the nodes in the CCR is more than in the LEACH. Adding new 100 nodes at round 400 increases the total energy in both protocols as shown in 27(a). The LEACH protocol returns to life for approximately 100 rounds (from 400 to 500) and dies once again. This shows that increasing the area affects the LEACH more than the CCR. Figure 27(b) shows that overflows in the LEACH occur as long as the network is alive (at the startup rounds and after adding new nodes).

Figure 27(c) shows that the total amount of data sent in the CCR is larger than in the LEACH. Figure 27(d) shows that the LEACH drops a lot of data while running (at the startup rounds and after adding new nodes).

5) SCENARIO 5

As experienced in the previous scenarios, the LEACH protocol dies faster than the CCR. This is also the case in this scenario, as shown in Figure 28. The number of rounds in which the LEACH is up and running has decreased to less than 10 rounds, while the CCR continues till round 500.

6) SCENARIO 6

Figure 29 shows that increasing the area decreases the lifetime of the LEACH protocol even after adding new nodes. Overall, the CCR protocol outperforms the LEACH protocol in all the previous scenarios. This performance increases with the increase of the number of nodes and the network area.

VI. CONCLUSION AND FUTURE WORK

This paper proposed a novel protocol for mitigating congestion and clustering in WSN. The proposed CCR protocol consists of two main phases; the setup phase and the transmission phase. A small-setup phase is used in rounds other than first round to remove dead nodes and solve dead clusters problem. The CCR protocol is characterized by the following features: (1) Low Overhead,(2) Load Distribution (3) Stability, (4) Reliability, (5) scalability and (6) Fault tolerance. Experimental results show that the proposed CCR protocol improves the performance of the network compared with the LEACH protocol, as it increases the network lifetime, does not suffer from any data overflow, and increases the number of the packets transmitted in each round. The stability of the proposed protocol, as the network area increases, is also proved.

As future work, nodes in the CCR protocol will use GPS to reduce the energy used for knowing the distance between nodes, and easily establish setup, small setup phases and routing tables. Also in future work, the CCR protocol will transmit data only when it changed not all data.

REFERENCES

[1] P. Sa, M. Sahoo, M. Murugappan, Y. Wu, and B. Majhi, "A new two-dimensional mesh topology with optical interlinks," in *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications* (Advances in Intelligent Systems and Computing), vol. 719. Singapore: Springer, 2017, pp. 437–442.

[2] A. Y. Haikal, M. Badawy, and H. A. Ali, "Towards Internet QoS provisioning based on generic distributed QoS adaptive routing engine," *Sci. World J.*, vol. 2014, Sep. 2014, Art. no. 694847.

[3] M. M. Badawy, Z. H. Ali, and H. A. Ali, *QoS Provisioning Framework for Service-Oriented Internet of Things (IoT)* (Cluster Computing). New York, NY, USA: Springer, 2019, pp. 1–17.

[4] A. Ladas, G. C. Deepak, N. Pavlatos, and C. Politis, "A selective multipath routing protocol for ubiquitous networks," *Ad Hoc Netw.*, vol. 77, pp. 95–107, Aug. 2018.

[5] H. Huang, A. V. Savkin, M. Ding, and C. Huang, "Mobile robots in wireless sensor networks: A survey on tasks," *Comput. Netw.*, vol. 148, pp. 1–19, Jan. 2018.

[6] K. Singh, K. Singh, L. H. Son, and A. Aziz, "Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm," *Comput. Netw.*, vol. 138, pp. 90–107, Jun. 2018.

[7] M. D. Maltz, J. Broch, D. Johnson, Y.-C. Hu, and J. A. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. 4th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, vol. 114, Oct. 1998, p. 119.

[8] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2000, pp. 255–265.

[9] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: A survey on recent developments and potential synergies," *J. Supercomput.*, vol. 68, no. 1, pp. 1–48, 2014.

[10] P. S. Shah, N. N. Patel, D. M. Patel, D. P. Patel, and R. H. Jhaveri, "Recent research in wireless sensor networks: A trend analysis," in *Information and Communication Technology for Sustainable Development*. Singapore: Springer, 2018, pp. 87–95.

[11] S. Jabbar, M. Ahmad, K. R. Malik, S. Khalid, J. Chaudhry, and O. Aldabbas, "Designing an energy-aware mechanism for lifetime improvement of wireless sensor networks: A comprehensive study," *Mobile Netw. Appl.*, vol. 23, no. 3, pp. 432–445, Jun. 2018.

[12] A. Ghaffari, "Congestion control mechanisms in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 52, pp. 101–115, Jun. 2015.

[13] A. Dumka, S. K. Chaurasiya, A. Biswas, and H. L. Mandoria, *A Complete Guide to Wireless Sensor Networks: From Inception to Current Trends*. Boca Raton, FL, USA: CRC Press, 2019. [Online]. Available: https://books.google.com.sa/books?id=pwadDwAAQBAJ

[14] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[15] W. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, and A. C. Smith, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2000.

[16] V. Loscri, G. Morabito, and S. Marano, "A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH)," in *Proc. IEEE Veh. Technol. Conf.*, Sep. 2005, vol. 62, no. 3, pp. 1809–1813.

[17] L. Lijun, W. Hongtao, and C. Peng, "Discuss in round rotation policy of hierarchical route in wireless sensor networks," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2006, pp. 1–5.

[18] F. Xiangning and S. Yulin, "Improvement on LEACH protocol of wireless sensor network," in *Proc. Int. Conf. Sensor Technol. Appl.*, Oct. 2007, pp. 260–264.

[19] H. Junping, J. Yuhui, and D. Liang, "A time-based cluster-head selection algorithm for LEACH," in *Proc. IEEE Symp. Comput. Commun.*, Jul. 2008, pp. 1172–1176.

[20] M. S. Ali, T. Dey, and R. Biswas, "ALEACH: Advanced LEACH routing protocol for wireless microsensor networks," in *Proc. Int. Conf. Elect. Comput. Eng.*, Dec. 2008, pp. 909–914.

[21] W. Wang, Q. Wang, W. Luo, M. Sheng, W. Wu, and L. Hao, "LEACH-H: An improved routing protocol for collaborative sensing networks," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, Nov. 2009, pp. 1–5.

[22] G. Yi, S. Guiling, L. Weixiang, and P. Yong, "Recluster-LEACH: A recluster control algorithm based on density for wireless sensor network," in *Proc. 2nd Int. Conf. Power Electron. Intell. Transp. Syst. (PEITS)*, vol. 3, Dec. 2009, pp. 198–202.

[23] A. Yektaparast, F.-H. Nabavi, and A. Sarmast, "An improvement on LEACH protocol (Cell-LEACH)," in *Proc. 14th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2012, pp. 992–996.

[24] G. N. Basavaraj and C. D. Jaidhar, "H-LEACH protocol with modified cluster head selection for WSN," in *Proc. Int. Conf. Smart Technol. Smart Nation*, Aug. 2017, pp. 30–33.

[25] A. Razaque, M. Abdulgader, C. Joshi, F. Amsaad, and M. Chauhan, "P-LEACH: Energy efficient routing protocol for wireless sensor networks," in *Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT)*, Apr. 2016, pp. 1–5.

[26] E. Alnawafa and I. Marghescu, "DMHT-LEACH: Dynamic multi-hop technique for wireless sensor networks," in *Proc. Int. Symp. Signals, Circuits Syst. (ISSCS)*, Jul. 2017, pp. 1–4.

[27] A. O. A. Salem and N. Shudifat, "Enhanced LEACH protocol for increasing a lifetime of WSNs," in *Personal and Ubiquitous Computing*. London, U.K.: Springer, 2019, pp. 1–7.

[28] G. K. Nigam and C. Dabas, "ESO-LEACH: PSO based energy efficient clustering in LEACH," *J. King Saud Univ.-Comput. Inf. Sci.*, to be published.

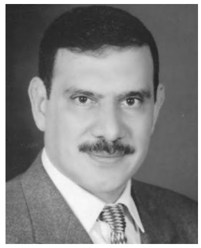
[29] M. Guray, N. Marriwala, and N. Ahmed, "Hybrid leach based cluster head election in wireless sensor networks," in *Proc. 4th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Sep. 2017, pp. 86–90.

[30] M. Bala and L. Awasthi, "Proficient D-HEED protocol for maximizing the lifetime of WSN and comparative performance investigations with various deployment strategies," *Int. J. Adv. Sci. Technol.*, vol. 45, pp. 107–124, Aug. 2012.
[31] M. Elshrkawey, S. M. Elsherif, and M. E. Wahed, "An enhancement approach for reducing the energy consumption in wireless sensor networks," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 30, no. 2, pp. 259–267, Apr. 2018.
[32] A. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," in *Proc. IPDPS*, vol. 1, Apr. 2001, pp. 2009–2015.
[33] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proc. IEEE Aerosp. Conf.*, vol. 3. Mar. 2002, p. 3.
[34] L. Buttyán and P. Schaffer, "Position-based aggregator node election in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 6, no. 1, Jul. 2010, Art. no. 679205.
[35] E. B. Hamida and G. Chelius, "A line-based data dissemination protocol for wireless sensor networks with mobile sink," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 2201–2205.
[36] S. Al-Sodairi and R. Ouni, "Reliable and energy-efficient multi-hop LEACH-based clustering protocol for wireless sensor networks," *Sustain. Comput., Inform. Syst.*, vol. 20, pp. 1–13, Dec. 2018.

MOHAMMED FARSI received the B.Sc. degree from King Abdulaziz University, Saudi Arabia, the M.Sc. degree in computer science from Newcastle University, U.K., and the Ph.D. degree in smart applications from the School of Engineering and Computing Sciences, Durham University, U.K. He is currently the Vice Dean of the College of Business Administration, Taibah University, Yanbu, Saudi Arabia, where he is also an Assistant Professor with the Information System Department, College of Computer Science and Engineering. His major research interests include artificial intelligence systems, cloud computing, database, interactive applications, and the Internet of Things.

MAHMOUD BADAWY received the B.Sc., M.Sc., Ph.D. degrees in computer engineering from the Faculty of Engineering, Mansoura University, Egypt, in 2004, 2009, and, 2014, respectively, where he is currently an Assistant Professor in computer engineering and system with the Faculty of Engineering. His research interests include (but are not limited to) computer networks, big data, deep learning, QoS, WSN, and distributed databases.

MONA MOUSTAFA received the M.Sc. degree in computer engineering from the Faculty of Engineering, Mansoura University, Egypt. Her research interests include computer networks, WSN, QoS, and distributed databases.



HESHAM ARAFAT ALI received the B.Sc. degree in electronics engineering, M.Sc., and Ph.D. degrees in computer Engineering and control from the Faculty of Engineering, Mansoura University, in 1986, 1991, and 1997, respectively, where he is a professor in computer engineering and systems and an associate professor in information systems and computer engineering. He was an Assistant Professor with the Faculty of Computer Science, University of Mansoura, from 1997 to 1999. From January 2000 to September 2001, he was a Visiting Professor with the Department of Computer Science, University of Connecticut. From 2002 to 2004, he was the Vice Dean of student affairs with the Faculty of Computer Science and Information, University of Mansoura. He has received the Highly Commended Award from the Emerald Literati Club 2002 for his research on network security. He is a Founder Member of the IEEE SMC Society Technical Committee on Enterprise Information Systems (EIS). He has many book chapters published by international press. He has about 150 published papers in international conferences and journals. His research interests include the areas of network security, mobile agent, network management, search engine, pattern recognition, distributed databases, and performance analysis. He has served as a Reviewer for many high-quality journals including the *Journal of Engineering* with Mansoura University.

YOUSRY ABDULAZEEM received the B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the Faculty of Engineering, Mansoura University, Egypt, in 2004, 2009, and, 2014, respectively. His M.Sc. subject was in software performance engineering and the Ph.D. subject was ranking distributed uncertain databases. In 2005, he joined the Computer Science Department, Misr Higher Institute for Computers and Commerce, Mansoura, Egypt, as a Teaching Assistant, and he became a Lecturer Assistant, in 2009. After receiving the Ph.D. degree, he joined the Computer Engineering Department, Misr Higher Institute for Engineering and Technology, Mansoura. His research interests include (but are not limited to) big data, deep learning, probabilistic database, query processing, distributed databases, software engineering, and software models.