International Journal of Innovative Computing, Information and Control
Volume **16**, Number **1**, February **2020**
ICIC International ©2020 ISSN 1349-4198
pp. 63–76

IMPROVE THE STABILITY OF THE INTERNET OF THINGS USING DYNAMIC LOAD BALANCING CLUSTERING

Shu-Ching Wang[1], Yu-Ling Lin[1], Mao-Lun Chiang[1,*] and Hsin-Hung Pan[2,*]

[1]Department of Information Management
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng, Taichung 41349, Taiwan
{ scwang; s10814901 }@cyut.edu.tw; *Corresponding author: mlchiang@cyut.edu.tw

[2]Department of M-Commerce and Multimedia Applications
Asia University
500 Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
*Corresponding author: vincentpan@asia.edu.tw

Received May 2019; revised September 2019

Abstract. *Nowadays, Internet of Things (IoT) presents enormous capabilities in terms of connecting everyday objects to be accessible from the Internet. Wireless Sensor Networks (WSNs) have been integrated into the perceived layer of IoT where Sensor Nodes (SNs) dynamically join the Internet and use them to collaborate and accomplish their tasks. Due to it the issues of information collection play an important role in the perceived layer of IoT. Energy efficiency is affected by the number of data transmissions from the SN to the Sink. However, the SN is limited by the energy resource, the memory, the computation, the communication capability, etc. Therefore, the hierarchical clustering topology has been proposed to prolong the lifetime of WSNs by decreasing the energy consumption of SNs. Unfortunately, the network topology is still unstable due to the fact that the workload of the Cluster Managers (CMs) is overloaded. To solve the aforementioned issues, a Dynamic Load Balance Clustering Mechanism (DLBCM) is proposed to balance the workload of CM and reduce the energy consumption of CM. Then, the lifetime of WSN will be prolonged.*
Keywords: Internet of Things, Perceived layer, Wireless sensor networks, Load balance

1. **Introduction.** Nowadays, the network has evolved from a human-oriented network pattern to an object-oriented IoT, forming a ubiquitous network environment [1]. The operation of the IoT is mainly divided into a three-layer structure. They are the perceived layer, network layer and application layer [2]. The main task of the perceived layer is to identify objects and to collect the information of various heterogeneous sensing data. The main function of network layer is to perform transmission and processing of information. The application layer combines the societal and real life usage requirements to create an intelligent application system.

In an IoT, devices can transfer data over network without use of any human interference with devices or other human being. With the help of IoT, objects can be controlled and sensed remotely across network, for integration of physical world into computer-based system. The integration results in more efficiency, accuracy, and economic benefits [3].

The WSNs are the important components of perceived layer under the IoT, being responsible for regional collecting and monitoring of data. A WSN is a wireless network consisting of spatially distributed autonomous devices using SNs in a wide range of applications in various domains. WSNs are integrated into the perceived layer of IoT, where

DOI: 10.24507/ijicic.16.01.63

SNs join the Internet dynamically, and use them to collaborate and accomplish their tasks. Currently, WSN has become a promoter technology for IoT applications, which extends the physical reach of monitoring capabilities. WSN itself has some constraints, such as limited energy availability, low memory capacity, and low processing speed, which is the main obstacle to designing an effective management protocol for WSN. In a WSN based IoT, the basic problem involves a mechanism to reduce node energy consumption, which will result in prolonging the life of the SN [4]. This is a very important consideration because the amount of energy consumed by transmitting a bit through the WSN is higher than the energy consumed by running a large number of microcontroller instructions.

Owing to the development of WSN it is progressing rapidly to become a popular network. Because of the ubiquity of IoT devices, the power consumption of IoT systems has become very important issues [5]. However, the capability of the SN is limited by the computation of the processor, the communication capability, energy resources, and the capability of memory, thus the network topology is easy to be destroyed. To achieve the energy-efficiency routing and prolong the lifetime of WSNs, data needs to be aggregated to organize the wireless environment [6] by a few neighbor nodes. The energy constraint is still a challenge even if the hardware technology has been enhanced [7]. Therefore, a Hierarchical Cluster-based Wireless Sensor Network (HCWSN) [8] is proposed to increase the stability and lifetime of the network by aggregation technology. In general, the Cluster Manager (CM) in HCWSN plays a key role to guarantee the communications of intra-cluster and inter-cluster. In general, each SN in the intra-cluster needs to send the data to the CM when it senses the event has occurred. After eliminating the redundant data, each CM needs to maintain and manage its cluster, such as announce the control messages. Besides, each CM needs to aggregate and transfer the simplified data to sink in the inter-cluster. The workloads of CM are heavy no matter intra-cluster or inter-cluster. As a result, the HCWSN topology needs to be reconstructed when the energy of the CM is completely exhausted. To solve the drawback of HCWSN, some power-aware algorithms [8] are proposed to prolong the lifetime of entire environment. However, the CM must be replaced when the temporal and burst communication events appear, which results in rapidly exhausting the energy of the CM. Then, the CM replacement frequently occurs resulting in extra energy consumption being necessary. Therefore, the network topology is still unstable and inefficient.

For improving the drawbacks of previous works [6-8], a concept is used in the proposed method, a part of the workload of CM can be shared actively to avoid the CM re-election or cluster reconfiguration when the CM is overloaded. The Backup Manager (BM) in our method needs to be invoked to share a heavy loading of the CM when the burst of incoming events occurs. It is because that the burst of incoming events is only to bring temporal and a large amount of data to aggregate, thus the loading of CM is overloading temporarily.

In this study, the Dynamic Load-Balance Clustering Mechanism (DLBCM) is proposed; then the lifetime of the network can be prolonged even if the incoming burst events occur. The proposed method not only considers the loading of the CM, but also monitors the energy consumption of the CM in each cluster to avoid the CM re-election and cluster reconfiguration frequently to keep the entire network topology more stable and efficient.

The remainder of the article is organized as follows. The related works of hierarchical clustering algorithms are presented in Section 2. The proposed method, DLBCM, is shown in Section 3. In Section 4, the results of simulation are illustrated. Finally, the conclusion is presented in Section 5.

2. The Past Research of the Classification of Clustering Mechanism in WSN.

In the perceived layer of IoT, billions of sensors with various ability of sensing, computing, communicating, and possibly actuating are deployed to provide specific services in various application domains. Therefore, data aggregation is an important concept to reduce energy consumption and amount of message exchange, irrespective of static or dynamic WSNs [9]. In general, a certain amount of sensors in the vicinity forms a cluster and elects a CM to aggregate the data in a WSN [10]. Subsequently, an effective energy-ware hierarchical clustering structure can be constructed to manage the data communication and aggregation.

There are two kinds of stand-on election methods of the CM, the distributed and centralized clustering methods [9]. The main difference between the distributed and centralized clustering methods is the method of CM election. The CMs can be elected by using the whole network information that is gathered by the sink or Base Station (BS) in a centralized clustering method. As a result, the CM elected by a centralized clustering method is more suitable and robust than the one elected by a distributed clustering method. However, the SNs must periodically report its related information (ex. energy state, position, and ability) to CMs that resulted in raising the overheads of the network.

The most famous centralized clustering algorithm, LEACH-C (LEACH-Centralized) clustering algorithm, needs to collect the location and energy of each SN to elect the high-energy power CM to manage all of the SNs [11]. However, there are large overheads of communication in the initial stage of CM election stage in the centralized clustering algorithm. Therefore, it is hard to adapt to the reality of the situation, especially for the energy-aware WSN.

The most famous distributed clustering algorithm of WSN, LEACH (Low Energy Adaptive Clustering), each SN has an opportunity to be elected as the CM when the random probability value Pr_i ($0 < Pr_i < 1$) is less than the threshold T_i (T_i: the energy threshold of sensor node i (SN_i)) in each round [12]. Further, the CM can be replaced during a period of time to reduce the entire average consumption of energy. However, the threshold T_i of the repetitive CMs can be set to zero to avoid repetitive CMs from being elected. Similarly, the threshold of each SN which is not yet elected is decreased during the round. This is to facilitate the SN which is not elected to have a greater opportunity to be a CM. The SN_i can be converted into a newly elected CM and announce the message to the neighbor nodes when $Pr_i < T_i$. Then, each SN should join and register to this new CM, which has a higher signal power when many announcement messages are received. However, a large number of broadcasting and redundant messages need to be exchanged in an initial stage, which results in problems of broadcasting storm and packet collision [12]. Similarly, the extra energy consumption of each node is necessary when the topology is reconfigured in every round of the CM election process. Therefore, the network topology is unstable and inefficient due to the fact that the energy consumption of the topology reconfiguration is redundant and frequent.

The Simple Load-Balance Clustering Mechanism (SLBCM) is another popular distributed clustering algorithm [13], which uses the actual residual energy, the degree of processor utilization, the communication bandwidth of the node, and the distance to the centre of the cluster as the criteria to elect the adaptive CM. Therefore, the SLBCM can obtain the best result and more adapt to the reality of the energy-aware WSN than others. One advantage of the SLBCM is the ability to reduce the large energy consumption in the initialization of network construction by the *countdown mechanism*. Besides, the SLBCM can also monitor and balance the loading of the CM by the BM actively to avoid topology reconfiguration when the energy of the CM is exhausted and crashed. However, a critical defect is easy to occur in SLBCM. It is because that the CMs of SLBCM are easy to be replaced frequently when the temporal and burst communication events appear, which results in rapidly exhausting the energy of the CMs. Then, the CM replacement is often to be invoked resulting in extra energy consumption being necessary. Therefore, the network topology is still unstable.

For improving the related hierarchical clustering algorithms, the Dynamic Load-Balance Clustering Mechanism (DLBCM) is proposed to replace the CM at an opportune moment to prolong the lifetime of the network and provide a stable network topology even if the temporal and burst workloads occur.

3. **The Proposed Method.** The objectives of the proposed DLBCM include reducing the energy consumption and loading of the CM, enhancing the stability and efficiency of the network topology, and to prolonging the lifetime of the network. In this research, the location of each SN can be assumed to be found through the position sensor. Similarly, the changeability of the WSN needs to be considered in any time, such as the mobility of SN. Therefore, the influence of the incoming and disappearing SNs on the network will be resolved through this study. Finally, each SN communicates with each other to collect the data or exchange the ability value synchronously in HCWSN environment.

The proposed DLBCM is divided into three phases, *the initial phase*, *the active phase*, and *the inactive phase*. Four mechanisms are proposed in the following phases, including *the countdown mechanism*, *election and clustering mechanism*, *load monitor mechanism*, and *load evacuated mechanism*. In general, the nodes can be classified as Sensor Node (SN), Cluster Manager (CM), and Backup Manager (BM) in this study. At first, the SN with the best ability, such as the residual energy, and the processor utilization, can be elected as the CM in the *initial phase*. Subsequently, the reminder SNs with high ability can be assigned as the BM. The main jobs of the CM include managing the members of its cluster and forwarding the messages of inter-communication to other CMs.

As a result, the energy consumption of the CM is heavy and the cluster reconfiguration is frequently recurring owing to the higher loading of the CM in comparison to other SNs. To solve the problem above, the second phase, the *active phase*, the loading of CM can be monitored and shared to the BM to reduce the workload of the CM. Namely, the job of the CM can be taken over by the BM in our proposed algorithm when the CM is overloaded. In final phase, the *inactive phase*, the exhausted CM can be replaced by the BM actively to avoid communication disruptions. Subsequently, the details of the three phases of DLBCM are described as follows.

3.1. **The initial phase of DLBCM.** At the initial phase, no CM has been elected. Every SN of network will produce a value by itself randomly, this value will minus one after every unit time interval. In other words, the *election and clustering mechanism* can be invoked to organize the cluster and elect the CM by using the *countdown mechanism* in the initial phase. The advantage of the *countdown mechanism* is that it helps to avoid exchanging a large number of broadcast messages when the election of the CM and network construction is invoked. When the value of some SN is to be zero, its own relevant information such as the actual residual energy (e), the processor utilization (u), the communication bandwidth of node (c), and the distance to the center of the cluster (d) will be transformed into a capacity value (p) according to Equation (1) in this research, which will be delivered to the nodes surrounded as the basis of CM election, while the weight values w_1, w_2, w_3 and w_4 will be defined in accordance with the importance of each item of factors for meeting the various needs. However, the weight values w_1, w_2, w_3, and w_4 can be re-defined in accordance with the importance of each item in different specific applications. Subsequently, the appropriate CMs and cluster structures can be constructed in the initial phase.

$$p = (e) * w_1 + (1/u) * w_2 + (c) * w_3 + (1/d) * w_4, \quad \sum_{i=1}^{4} w_i = 1 \tag{1}$$

The advantages of this method are easily for reducing the time of election of CM besides reducing the transmission of information. The reason of random value to be down counted is as follows: Because all SNs must compare capacity value with the surrounding SNs to produce the CM at the beginning, if all SNs execute this operation simultaneously, a large number of transmissions will be suddenly produced to result in the paralysis of network operation. Therefore, the random count value can disperse the action of delivery and comparison to reduce the load of network at the beginning of comparison. What is more, this research set up the comparison that the transmission distance (delay time) between two SNs has to be lower than some value for getting the optimum result of information transmission in every cluster.

When a new CM is decided, the ID of CM is broadcasting to other SNs in its cluster. Next, the reminder SN with the highest capability will be elected as BM in the same cluster. After the BM is decided, the BM broadcasts the message "I am BM" to all the SNs in its cluster. Moreover, the BM can convert into the new CM and take over the cluster when the CM fails (such as crash, silence or energy exhausted). Subsequently, the *active phase* is shown in the following section.

3.2. **The active phase of DLBCM.** Various factors are used and the respective weights are collected to elect the suitable CM to manage the cluster. However, the workloads of a CM are more than other SNs, such communication in an inter-cluster, data aggregation, and communication with sink. Therefore, the CM may be replaced by BM when the CM is overloaded. Moreover, a new CM may need to be elected again when a succeeding CM is exhausted. Therefore, the cluster is still unstable and inefficient. To solve the problems above, two kinds of states must be considered.

1) CM failed (such as crash, silence, or energy exhausted): the BM can be invoked to take over the CM.
2) CM overloaded: the workload of CM is heavy or the ability of CM is less than the threshold. Based on the *load monitor mechanism* and *load evacuated mechanism*, the CM needs to be adjusted by different situations.
   i. New_L_t (the new threshold of loading ability of CM_i) $< L_i$ (the loading ability of CM_i) $< L_t$ (the threshold of loading ability): The BM is authorized to handle the following jobs of data transmission and data aggregation.
   ii. $L_i < New\_L_t$: The BM is invoked to take over the job of CM.

According to the aforementioned reasoning, the *load monitor mechanism* and *load evacuated mechanism* are proposed to prolong the lifetime of the CM and increase the network stability in *active phase*. The *load monitor mechanism* is proposed to estimate and monitor the loading of the CM to avoid the CM replacement is invoked frequently in this phase. Subsequently, the latter mechanism, the *load evacuated mechanism* is invoked to prevent the sudden energy exhaustion of the CM when the loading of the CM is detected as overloaded.

(A) **The Load Monitor Mechanism of Active Phase:** After the CM is elected, the loading ability (L_i) can be recomputed to evaluate the degree of loading by Equation (2). This Equation (2) is improved from Equation (1) and including the actual residual energy (e), the processor utilization (u), and the communication bandwidth of node (c), but not the distance to the centre of the cluster (d) due to the factor

(d) is irrelevant to the loading ability of the CM. Thus, the loading ability (L_i) must only be computed by factors e, u, and c. Subsequently, the threshold of loading ability (L_t) is computed by Equation (3), which includes the same factors as Equation (2). The parameters $t_1 \sim t_2$ of Equation (3) can be separately adjusted to suit the actual environment in accordance with the requirement of the user. In our case, the parameters can be set to 1/3 separately. Besides, the CM can retire and be actively replaced with the BM in this paper when the current CM is becoming inactive ($L_i < L_t$).

$$L_i = (e) * w_1 + (1/u) * w_2 + (c) * w_3, \quad \sum_{i=1}^{3} w_i \leq 1 \tag{2}$$

$$L_t = (e) * t_1 + (1/u) * t_2 + (c) * t_3, \quad L_t \leq L, \quad \sum_{i=1}^{3} t_i \leq \sum_{i=1}^{3} w_i \leq 1 \tag{3}$$

In general, the BM is invoked to replace the CM, when the CM is overloaded. However, the sudden appearance of a burst of incoming events needs to be considered to avoid redundant CM replacement. It is because that the new CM may be temporarily overloaded due to the burst of incoming events. Therefore, the loading ability of the retired CM, which has been replaced, can be recovered quickly and kept steady when the burst of incoming events are accomplished. Namely, the retired CM could still have the best ability in its cluster and the CM replacement could have been invoked too early. Specifically, the CM replacement could happen again and the costs involved would be extraneous expenses. For calculating the suitable time to invoke the CM replacement, the second mechanism, the *load evacuated mechanism*, has been proposed in the following section.

(B) **<u>The Load Evacuated Mechanism of Active Phase:</u>** In traditional methods [5], the CM with a low L_i needs to be replaced when $L_i \leq L_t$. However, the CM replacement is easy to be invoked frequently, thereby making the network topology unstable. For solving problem above, the concept of *slow start* [14] is used in the *load evacuated mechanism* to prolong the entire lifetime of the CM and avoid extraneous expenses. Besides, the burst of incoming events bring a large amount of data to aggregate. Thus, the BM is invoked to share the job of the CM on demand. To avoid data inconsistency, the BM only takes over the subsequent works (T_{i+1}: the time $i+1$), such as unfinished data aggregation. The workload distributions between the CM and BM are shown in Figure 1. Based on the load evacuated mechanism, the CM announces the message, "*I am busy*", to all the members of its cluster when the loading ability is less than the threshold of the loading ability ($L_i \leq L_t$). Subsequently, the BM is authorized to handle the following jobs of data transmission and data aggregation, such as data transmission T_{i+1}. However, the job of data transmission T_i is still to be handled by the original CM, as T_i comes prior to the announced message, "*I am busy*". The BM is yet to execute the unfinished work and report to the CM. The job retrieval of the CM is shown in Figure 2. However, the loading ability of the CM decreases with time. The CM is then replaced by the BM and subsequent works are implemented by the BM. Unfortunately, the network reconstruction is necessary when the loading ability of the BM is still less than L_t. It suggests that all the nodes of this cluster are unsuited to maintain the job of the CM and a new cluster is required. Consequently, the opportune moment of CM replacement would be a critical issue, which could influence the lifetime of network. The opportune moment

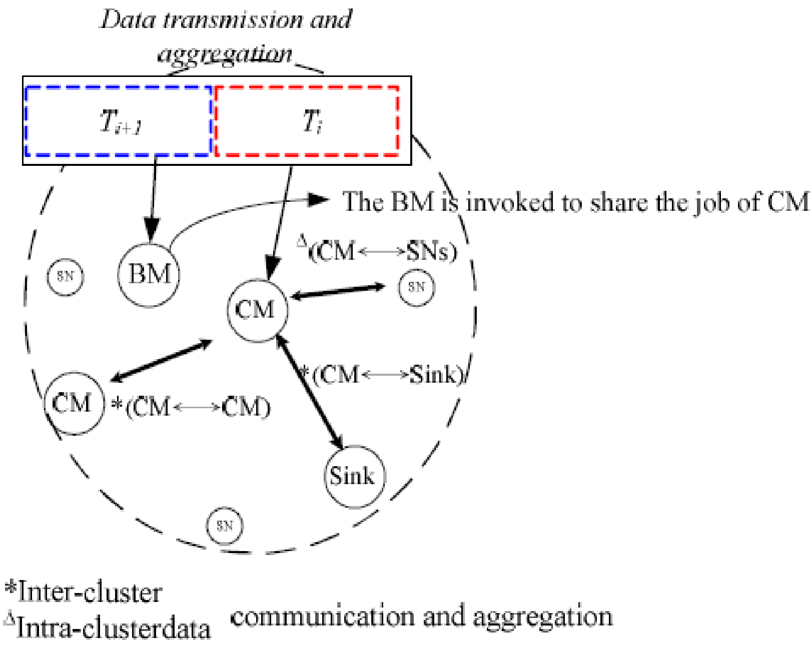

FIGURE 1. The distributions of workload between CM and BM

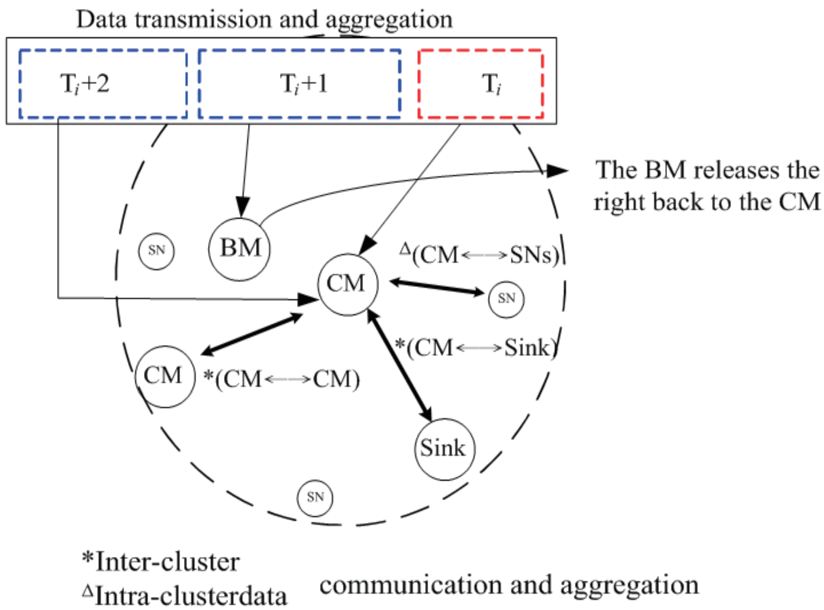

FIGURE 2. The job retrieval of CM

of CM replacement can be decided based on the design of the threshold of the loading ability.

(C) **The Adjustment of the Threshold of the Loading Ability:** In general, the retired CM can be elected as CM again when the loading ability (L_i) is more than the threshold of the loading ability (L_t). It is because the loading of the processor utilization (u) and communication bandwidth of the node (c) is heavy to result in the $L_i < L_t$ when the burst incoming events disappear. Therefore, the ping-pong effect [15] may be generated when the L_i approximated to the L_t. To avoid the CM replacement again, the threshold of the loading ability needs to be adjusted to suit the current loading ability of the CM. Based on the aforementioned reasoning, and the concept of *slow start* is used in the *active phase*. The new threshold of the loading ability (New_L_t) can be redesigned for a reduction of 25% during each phase through Equation (4) when the *loading evacuated mechanism* is invoked every time. It is because that the energy of SN is more limited, hence the concept of slow start technique is used to set the reduction ratio as 25%. Therefore, the energy factor of New_L_t needs to multiply 75% with time and adapted to the real situation. The frequency of CM replacement can be reduced efficiently and the ping-pong effect can also be avoided. Furthermore, the CM can focus on the jobs of its cluster and not to waste extra energy to execute the CM replacement. Subsequently, the pseudo code of the adjustment of the threshold of the loading ability is shown in Figure 3. Besides, the complexity of this function is $O(N^2)$ when the CM broadcasts the messages to ask the help or retrieve the control right. Only BM needs to be invoked to take over the job of CM or return the control right; hence, the complexity can be reduced.

$$New\_L_t = (e) * t_1 * 0.75 + (1/u) * t_2 + (c) * t_3 \tag{4}$$

```
Function Load Evacuated Mechanism (C_j)
    L_t = (e) * t_1 + (1/u) * t_2 + (c) * t_3; //Compute the threshold of loading ability.
    L_i = (e) * w_1 + (1/u) * w_2 + (c) * w_3; //Compute the loading ability of CM.
    if L_i ≤ L_t then
        the CM broadcasts the message "I am busy" to all;
        the BM takes over the following jobs and processes the data aggregation;
        if L_i of CM > L_t then
            t_1 = t_1 * 0.75; //adjust L_t after CM recovery during a period of time;
            New_L_t = (e) * t_1 + (1/u) * t_2 + (c) * t_3;
            the CM broadcasts the message "I am ready" to all SNs;
            the CM retrieves the right from the BM and implements the following job.
    else
        the L_i of the CM is monitored continuously.
End function
```

FIGURE 3. The pseudo code of the load evacuated mechanism

3.3. **The inactive phase of DLBCM.** In general, the *load evacuated mechanism* is invoked when $L_i \leq L_t$. However, the loading ability of the CM (L_i) is still less than the L_t even if the loading is shared by the BM. The CM with a low ability needs to be replaced by the BM actively to avoid communication disruptions, which could result in an unstable network. In addition, the CM may be inactive and disappear unexpectedly. Thus, the BM needs to be invoked to take over the job of the CM at any time. However, the cluster will be destroyed and the *initial phase* will be invoked for constructing a new cluster again, when the BM is silent or less than L_t.

When the failure of calling communication between SN_i and CM in cluster j happens, SN tries to call the CM until the threshold *TE* (the time of try and error). If CM is still alive then CM keeps working, else BM is invoked to become the new CM. When the BM is woken up as the new CM then the BM takes over the job of the old CM and a new BM is elected by new CM. In addition an announcement is sent to all SNs. If the BM cannot be woken up, the initial phase is invoked again. The details of the *inactive phase* of DLBCM are shown in Figure 4.

According to the description above, the CM replacement will not to be invoked frequently by *load monitor mechanism* even if the burst communication events appear. Besides,

```
Inactive Phase (C_j)
    if the CM is silence in cluster j then //The failure of calling communication
                                             between SN_i and CM in cluster j.
      for (f = 1; f < TE; f++) //Try to call the CM until TE.
        if CM is still alive then
          CM keeps working;
          break.
        else
          the BM is invoked;
          if the BM is woken up then
            CM = BM; //The BM takes over the job of the CM;
            the new CM assigns a new BM and sends an announcement
            to all SNs;
            break.
          else
            the initial phase is invoked again;
            break.
      next.
```

FIGURE 4. The pseudo code of the inactive phase

the loading of CM can be detected and shared by BM when the $L_i > L_t$. To avoid the ping-pong effect, the L_t can be adjusted by New_L_t to adapt to the real situation in *load evacuated mechanism*. Finally, the BM can take over the job of CM actively when the CM may be inactive and disappear unexpectedly in *inactive phase*. As a result, the DLBCM can prolong the lifetime of the CM and increase the network stability in WSNs. The same as function *load evacuated mechanism*, the BM needs to be woken up when the CM is silence. Therefore, the complexity can be reduced to $O(N^2)$.

4. **Experimental Results.** In this section, the proposed DLBCM will be compared with the popular clustering methods, LEACH-C [11], LEACH [12] and SLBCM [13]. In the present experimental environment, NS2 [16] is used to generate 50 and 100 SNs under a 100 m $*$ 100 m perceived layer of IoT, respectively. In addition, the experiments are repeatedly executed about 30 times to verify whether the proposed DLBCM method can efficiently prolong the entire lifetime of network.

4.1. **The assumptions of the experimental environment.** In order to verify that the method proposed in this study can improve the stability of IoT effectively, the parameters used in the experiment are based on the settings of the related researches [3,11,13]. The parameters of this experiment include network status, node characters, and node energy, as shown in Table 1. All parameters are designed to verify under the same environment. Besides, the weight values can be set to 0.25 to adapt to general environment in this study and the characters of the node assumed as follows.

1) Each SN consumes its energy power during the execution time until the energy gets exhausted. Each SN cannot recharge when the location deployment is set at random.
2) Each SN has a unique identifier (ID) and knows its position by the position device.
3) The transmission of data can be generated and transferred continuously.

4.2. **The design of the experiment.** Based on the previous works [3], the lifetime of the network, the amount of data aggregation, and the energy consumption are critical

Table 1. The parameters of the experimental environment

Items	**Parameters**	**Values**
Network status	Area	$100 * 100$ m^2
	Number of nodes	50 and 100 nodes
	Location of Sink	(0, 0)
	Cluster range	30 m
	Size of data packet	500 bytes
	Size of announcement packet	25 bytes
Node characters	Initial energy	1 J/battery
	The process ability of node	8 MHz
	Bandwidth	250 kbps
Node energy	E_{elec}	50 nJ/bit
	E_{friss_amp}	0.6185 nJ/bit/m^2
	$E_{two\text{-}ray\_amp}$	0.13365 pJ/bit/m^4

evaluate factors. Therefore, the comparisons of the proposed DLBCM with LEACH-C LEACH, and SLBCM are shown in our experiments. In the first experiment, the lifetime of the network and the amount of data aggregation can be computed to verify the improvement on the stability and efficiency of the network. Furthermore, a detailed and precise discussion of energy consumption is mentioned in the second experiment to ensure that the DLBCM can reduce the excess energy consumption by using the *load evacuated mechanism*, in comparison to the other algorithms.

4.3. **The results of the first experiment.** In the first experiment, the algorithms, LEACH-C, LEACH, SLBCM, and DLBCM are simulated 30 times, as shown in Figures 5, 6, 7, and 8. Besides, the averages are taken under the 50 and 100 SNs, respectively. In Figures 5 and 6, the vertical axis indicates the number of the survivor nodes, while the horizontal axis indicates the time of the unit (each unit is 10 seconds). Several observations in Figures 5 and 6 demonstrate that the lifetimes of the network in SLBCM and DLBCM are longer than LEACH-C and LEACH. The lifetimes of the network in LEACH-C and LEACH are nearly exhausted in 500 seconds. This is because the cluster reconfiguration is frequently occurring and consuming 10 times of initial energy (about 10 J/battery) under the LEACH-C and LEACH. Therefore, it is easy for the cluster reconfiguration to exhaust the energy power of each SN. However, the lifetime of the network in a centralized LEACH-C is more stable in comparison to that of LEACH. The LEACH-C can elect a proper number of CMs. Thus, the advantage is more obvious under the environment with a large number of SNs.

In contrast to the LEACH-C and LEACH, the lifetime of the network in SLBCM and DLBCM can be prolonged over 10000 seconds. These algorithms operate better since they use the BM to take over the job from the CM in place reconstructing the cluster in any period of time. The results between the SLBCM and DLBCM were compared with the obtained results showing that the lifetime of the network in the DLBCM is longer than the one in the SLBCM. The *load evacuated mechanism* of DLBCM can be invoked to avoid the CM being replaced directly by the BM, when the loading ability of the CM is lower than the threshold of the load ability ($L_i \leq L_t$). This is because energy consumption is huge and rapid when the CM replacement is frequently occurring.

However, it can be seen that the initial period of lifetime of the network in SLBCM can obtain the better result than DLBCM in Figures 5 and 6. This is because the DLBCM

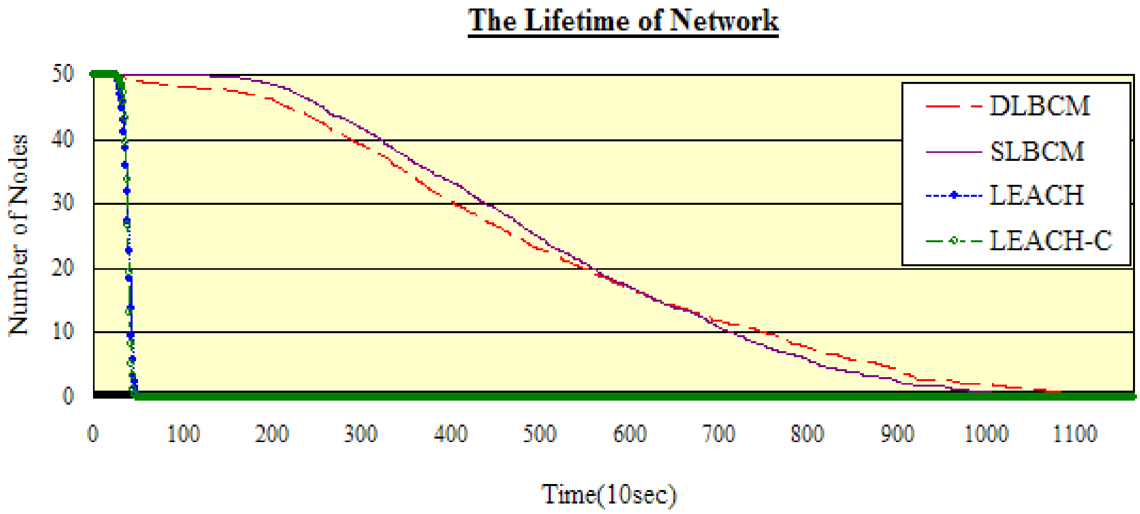

FIGURE 5. The number of survivor nodes under 50 nodes

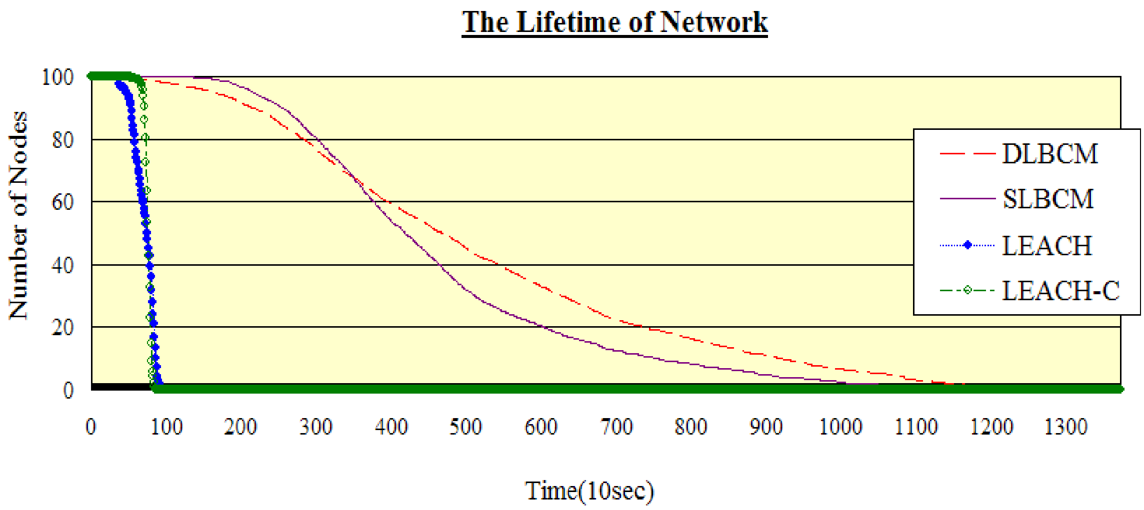

FIGURE 6. The number of survivor nodes under 100 nodes

needs to consume some energy power to monitor and invoke the proposed mechanisms in this study. Besides, the amount of data aggregation of SLBCM is lower than one of DLBCM in Figure 7. It is a trade-off to consider the data aggregation and energy consumption. In the later period of time, the lifetime of the network in DLBCM is longer than that in SLBCM. Similarly, the result of DLBCM is better than one of SLBCM in 3500 s when the survivor nodes are approximate to 70 in Figure 6. Therefore, the advantage of DLBCM is obvious and efficient when a large number of WSNs exist.

The results of the average number of data aggregation are illustrated in Figure 7. Obviously, the SLBCM and DLBCM are better in comparison to others due to the fact that a large number of data can be collected. Additionally, the DLBCM can obtain a better result in comparison to SLBCM when the number of nodes is equal to 100. This is because that the loading of the CM is easy to be exhausted and the number of data in 100 SNs environment is hard to be aggregated. Thus, the *load evacuated mechanism* could

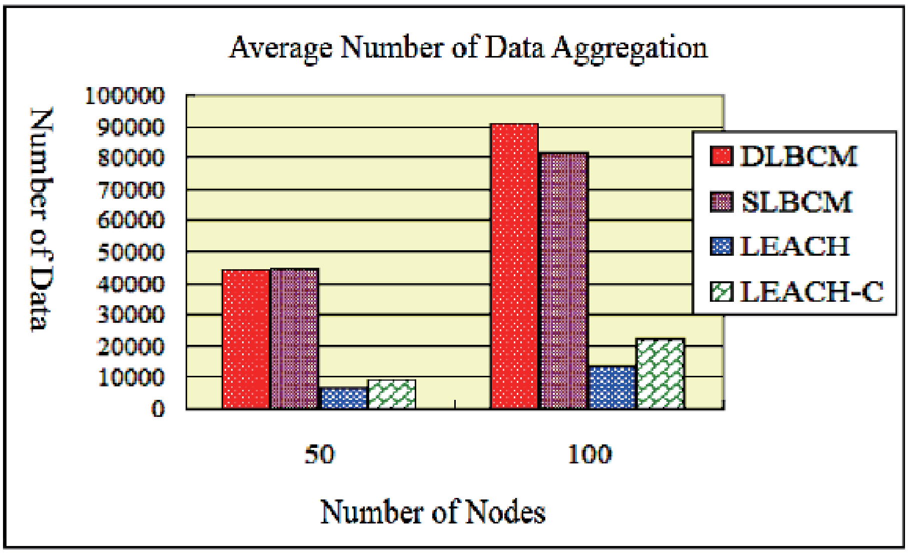

Figure 7. The results of the average number of data aggregation

Number of nodes	50	100
DLBCM	11013.33 seconds	9170.667 seconds
SLBCM	9465.333 seconds	8690.333 seconds

Figure 8. The average lifetime of the network between DLBCM and SLBCM

be useful to adjust to adapt to the real environment, a large number of SN environments especially. Namely, the threshold of the load ability (L_t) can be used to invoke the *load evacuated mechanism* depending on the workload and density of network. As a result, the average lifetime of the network in DLBCM is longer than SLBCM and the DLBCM is more adapted to the WSN of IoT, as shown in Figure 8.

4.4. **The results of the second experiment.** In this experiment, the *load evacuated mechanism* can be verified to ensure that the DLBCM can reduce the excess energy consumption of the CM, thereby prolonging the lifetime of the entire network. We averaged 30 times in this experiment to show the average residual energy of the CM and the BM under the 50 node and 100 node environments in Figures 9 and 10, respectively.

The energy consumption of the CM is more than that of the BM due to the fact that the CM needs to handle the management of the cluster, as shown in Figure 9. Besides, the average residual energy of the CM in the SLBCM is better than the one in DLBCM before 6000 seconds. This is because the CM replacement is invoked earlier in SLBCM. Thus, the energy consumption of the CM is less than that of DLBCM. However, the CM replacement is invoked too early and the BM is wasting extra energy in an initial period of time. Therefore, the average energies of the CM and the BM in the SLBCM are easy to be exhausted in a later period of time. In contrast to the SLBCM, the energy consumptions of the BM and the CM in the DLBCM are slower, and the average residual energy is the highest among all algorithms. Similarly, the advantages of DLBCM are more obvious when the number of nodes increases. The average residual energy in the DLBCM is higher than all the other methods after 4000 seconds; the result of detail is demonstrated in Figure 10.

Based on the aforementioned observations, the DLBCM can reduce the energy consumption of the CM, thereby prolonging the entire lifetime of the network. Besides, the

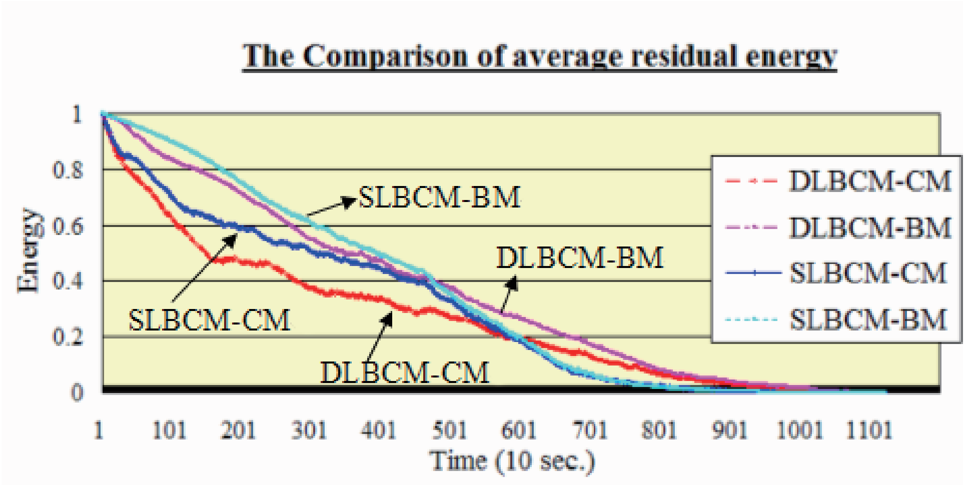

FIGURE 9. The comparison of average residual energy between CM and BM under 50 nodes

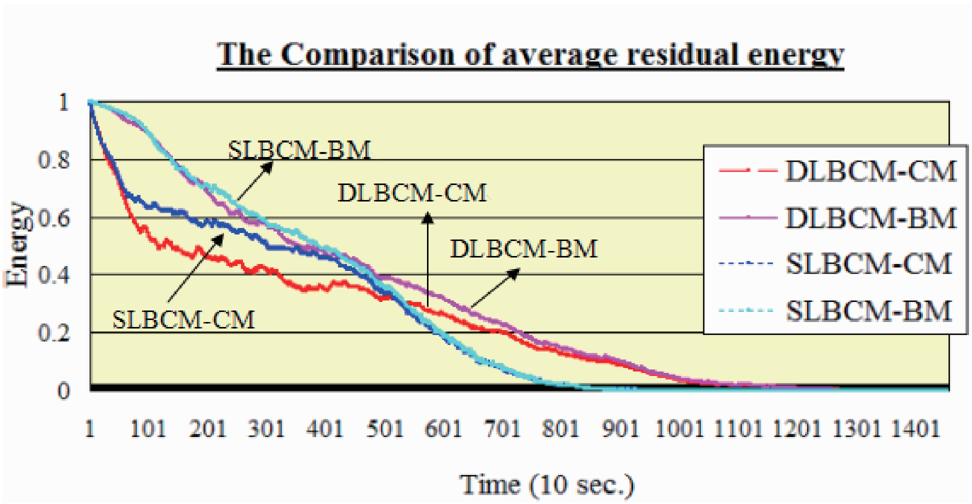

FIGURE 10. The comparison of average residual energy between CM and BM under 100 nodes

proposed DLBCM can balance the loading between the CM and the BM to make the network topology more stable. Finally, the average number of data aggregation can also obtain the highest amount of data among all the algorithms.

5. **Conclusion.** In the perceived layer of IoT, WSNs are very popular lately, but they are limited to energy and computing power [9]. As a result, some clustering algorithms are proposed to prolong the lifetime of the WSNs by decreasing the energy consumption of the SNs [3]. Unfortunately, the network topology remains unstable. This is because the CM replacement and network reconstruction are easy to occur frequently due to the overloading of CM. Therefore, the DLBCM is proposed to construct a stable and energy-aware network topology for prolonging the lifetime of the network. Besides, the average number of data aggregation of DLBCM can obtain the highest amount of data among all of algorithms. According to the results obtained in the aforementioned experiments from this study, the following goals can be achieved: 1) design a stable mechanism to form a hierarchal WSN topology; 2) balance the workload of the CM and reduce the energy consumption; 3) enhance the stability and performance of the whole network; and 4)

prolong the lifetime of the whole network. Thus, the proposed DLBCM is more efficient and useful to adapt to a large number of WSNs environment, especially.

In future, the distance and the speed of movement of SN in the perceived layer of IoT will be considered. When GPS is presented, the factors of distance and speed will be readily available. In addition, it will make the system more rigorous and more challenging.

Acknowledgment. This work was supported in part by the Ministry of Science and Technology MOST 107-2221-E-324-005MY3.

REFERENCES

[1] M. A. Khan and K. Salah, IoT security: Review, blockchain solutions, and open challenges, *Future Generation Computer Systems*, vol.82, pp.395-411, 2018.
[2] H. A. Khattak, M. A. Shah, S. Khan, I. Ali and M. Imran, Perception layer security in Internet of Things, *Future Generation Computer Systems*, vol.100, pp.144-164, 2019.
[3] M. Elappila, S. Chinara and D. R. Parhi, Survivable path routing in WSN for IoT applications, *Pervasive and Mobile Computing*, vol.43, pp.49-63, 2018.
[4] M. Lyu, J. Zhang, L. Liu, Y. Bo and F. Xiao, Discrete sliding mode control for a class of WSN based IoT with time-delay and nonlinear characteristic, *Journal of Internet Technology*, vol.20, no.2, pp.491-507, 2019.
[5] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand and A. H. Gandomi, Residual energy based cluster-head selection in WSNs for IoT application, *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2019.2897119, 2019.
[6] M. J. Neely, Energy-aware wireless scheduling with near-optimal backlog and convergence time tradeoffs, *IEEE/ACM Transactions on Networking*, vol.24, no.4, pp.2223-2236, 2016.
[7] S. Malhotra, S. K. Jain and V. Shrivastava, Improved performance using cluster based routing methods for wireless sensor network, *Journal of Telecommunication, Switching Systems and Networks*, vol.4, no.3, pp.5-11, 2018.
[8] P. Verma, S. Shaw, K. Mohanty, P. Richa, R. Sah and A. Mukherjee, A survey on hierarchical based routing protocols for wireless sensor network, *Proc. of the IEEE International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 2018.
[9] A. S. Rostami, M. Badkoobe, F. Mohanna, A. A. R. Hosseinabadi and A. K. Sangaiah, Survey on clustering in heterogeneous and homogeneous wireless sensor networks, *The Journal of Supercomputing*, vol.74, no.1, pp.277-323, 2018.
[10] M. A. Sayeed and R. Shree, Optimizing unmanned aerial vehicle assisted data collection in cluster based wireless sensor network, *ICIC Express Letters*, vol.13, no.5, pp.367-374, 2019.
[11] P. Garg and R. Rani, A survey on wireless sensor networks routing algorithms, *Proc. of the 16th International Conference on Formal Engineering Methods*, 2014.
[12] S. Xiao and Z. Xu, Reliable and energy efficient communication algorithm in hierarchical wireless sensor networks, *Wireless Personal Communications*, vol.95, no.3, pp.1891-1909, 2017.
[13] K. Q. Yan, S. C. Wang and Y. P. Tung, The framework for cluster manager election and clustering mechanism in heterogeneous wireless sensor networks, *Proc. of the 4th International Conference on Information Management*, 2008.
[14] C. Döpmann and T. Florian, CircuitStart: A slow start for multi-hop anonymity systems, *Proc. of the ACM SIGCOMM 2018 Conference on Posters and Demos*, pp.6-8, 2018.
[15] B. Naeem, R. Ngah and S. Z. M. Hashim, Reduction in ping-pong effect in heterogeneous networks using fuzzy logic, *Soft Computing*, vol.23, no.1, pp.269-283, 2019.
[16] The VINT Project, *NS Manual*, http://www.isi.edu/nsnam/ns/, 2005 (Access on 2019.01.15).