

## Research Article

# DACH: An Efficient and Reliable Way to Integrate WSN with IPv6

Dequan Yang and Qiao Guo

*School of Automation, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Dequan Yang, yangdequanbit@gmail.com

Received 27 June 2012; Revised 25 September 2012; Accepted 31 October 2012

Academic Editor: Chuan Foh

Copyright © 2012 D. Yang and Q. Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IPv6 has many advantages such as the massive amount of addresses, high security, and high robustness, which are beneficial for wireless sensor networks (WSNs). However, it is almost impossible to use IPv6 directly in WSN due to its huge energy consumption. This paper proposes a double adaptively clustering hierarchy (DACH) algorithm which enables using IPv6 in WSN in an efficient and reliable way. Firstly, we present a clustering method to adaptively divide the whole sensor network into clusters according to its energy consumption in the last round. Then we propose an adaptive cluster head selection algorithm which employs a strategy to choose the most suitable cluster heads; meantime, this selection algorithm is integrated into DACH. Finally, the complete framework is built between headers and their slave nodes based on IEEE 802.15.4, and IPv6 is used to connect the headers and the base stations. Experimental and simulation results demonstrate that the DACH algorithm has lower time and energy consumption. Moreover, it is more reliable and applicable than many other IP-based WSN algorithms.

## 1. Introduction

One of the most important techniques of this decade is wireless sensor networks (WSNs). In the last twenty years, interpersonal communication has become very popular with the booming internet technology. Similarly, with the development of WSNs [1], the same phenomenon will occur, and people will benefit a lot from this new information exchange technology. When WSN is as widely used as the internet, people can turn on their air conditioners at home when they are still on their way; the information of snow depth of every valley of Alps can be measured and collected by sensors and sent to people for making decisions about holiday skiing; any equipment of a city can send an alarm to the fire station automatically when the temperature is beyond the normal range, and so forth.

Without access to the internet, WSN is just a usual local network with its limited power. However, when IPv6 joins, WSN becomes magic and powerful, for IPv6 has a lot of advantages, such as massive addresses, high security, and good QoS service [2]. Since TCP/IP is limited with factors like too much energy cost and low battery frequent data transmission at the sensor nodes, IPv6-based WSN is more favorite for the researchers. However, for WSN, header

overhead problem in IPv6 is more serious than that in IPv4. Usually, the monitoring signal, control signal, and measured data of a sensor is no more than 10 bytes [3]. If IPv6 is introduced directly, the header overhead will consume more than 80% of the total energy and drastically shorten the lifetime of WSN.

As is shown in Figure 1, the header length of IPv6 is nearly one-third of the upper bound of the total frame length under IEEE 802.15.4, which takes up too much resource and energy consumption. To solve this problem, many researchers are devoted to finding efficacious ways to integrate WSN with IPv6.

To make WSN access to the internet, [4] proposes an all-IP scheme in which all the sensor nodes take IPv6 as the internet protocol. In this way, WSN can be integrated into the internet seamlessly, and WSN becomes part of the internet. However, the all-IP scheme often leads to a large energy consumption, and when the energy cost of all-IP WSN is too high to be tolerated, the whole network will be down.

In [5], a high energy efficiency strategy is employed to make WSN access to the internet. Although the lifetime of the nodes in [5] is ideal for applications in reality, the strategy is based on IPv4. Therefore, this strategy cannot be used in IPv6-based sensor networks.

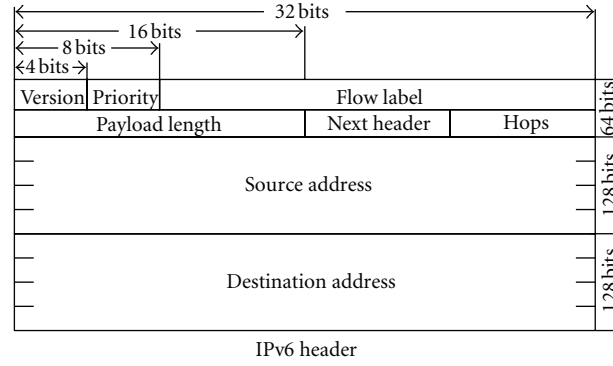


FIGURE 1: Structure of IPv6 header.

In [6, 7], the scholars try to build a data-centric IP-based WSN routing protocol, in which the algorithms have acceptable time and computational overhead. But these methods are not well adapted to the address-centric internet.

In [8], a mixed scheme is proposed, in which it combines agent-based IP WSN and TCP/IP-WSN to make WSN access to the internet. Theoretically, this method is perfect, but no details of this algorithm is given, and plus, no experiment results and even simulations are provided to support this theory.

In [9], the stack is modified to integrate IPv6 into sensor network. The simulation verifies its good performance. However, the plan is too complicated and difficult to deploy in reality.

Although there are a lot of studies on the solution for the application of compressed IPv6 to WSN, compressed IPv6 is also a high energy consumption protocol, which needs the support of inner-cluster communications. Hence, an algorithm based on inner-cluster connection for high energy efficiency WSN is needed to save more energy for the heads to support IPv6 communications.

In this paper, a double adaptive clustering hierarchy (DACH) algorithm is proposed to introduce IPv6 into WSN to make sensor networks able to access to the internet. Firstly, a clustering algorithm is designed to divide WSNs into several clusters adaptively according to the number of hops and energy consumption. Secondly, a cluster head selection algorithm is constructed to select clusters adaptively, which can further reduce the energy consumption. Finally, communications in cluster are based on IEEE 802.15.4, and one cluster connects with other clusters or with base stations through IPv6. The paper contributes the following:

- (1) a clustering algorithm which makes clustering adaptively and saves energy cost;
- (2) an adaptive cluster head selection algorithm which enables the network to choose headers with minimum energy consumption;
- (3) a reliable IPv6-based WSN framework which makes WSN access to the internet in a reliable way with high energy efficiency;

- (4) simulations and experiments are present to demonstrate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 reviews the problems of WSN clustering. In Section 3, the adaptive clustering algorithm for dividing WSN into groups is presented. In Section 4, a weighing mechanism is designed, which takes the parameters such as remaining capacity, number of hops, and cluster head history into consideration, to build a cluster head selection algorithm. The low energy consumption access to internet of WSN is achieved by a switching strategy between IEEE 802.15.4 and IPv6 in Section 5. In Section 6, simulations and experiments are presented and analyzed. And Section 7 is the summary of this paper.

## 2. Background

**2.1. Why Clustering?** Since the survival time of nodes in WSN is limited by the remaining capacity of their batteries, one of the most significant challenges in WSN design is how to save energy as much as possible [10].

Clustering is an important topology control method in WSN. It can significantly reduce average energy consumption of nodes and improve network throughput. In clustering routing protocols, sensors networks are usually divided into clusters. A cluster is a node set in which the nodes are connected together in a certain way. Each cluster contains a cluster head and a number of cluster members. Cluster heads of lower-level networks are the cluster members of higher-level networks. The highest level of cluster head works for communicating with base station [11]. The topology of cluster-based WSN is shown in Figure 2.

In clustering algorithm, the whole sensor network is divided into a number of subsets, which are connected with each other by the heads. In each cluster, the head is selected by a certain algorithm which can manage or control all the members of the cluster. In addition, the head should be responsible for inner-cluster information collection, data fusion, and intercluster data forwarding [12].

The advantages of clustering routing are as follows.

- (1) Most of the time, cluster members could turn off their communication models because the cluster

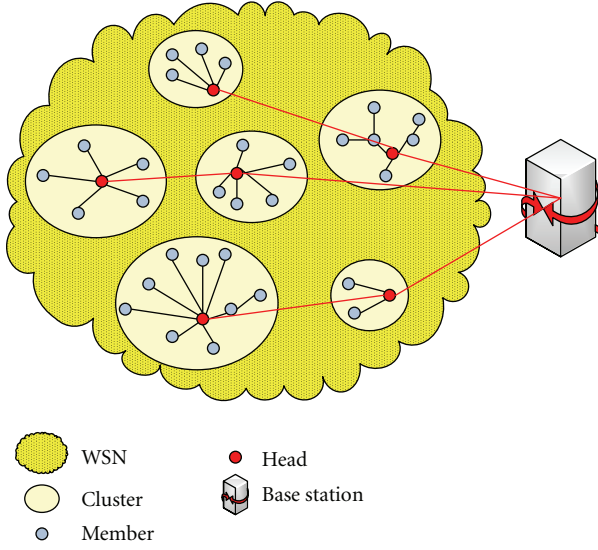


FIGURE 2: Topology of the cluster-based WSN.

heads constitute a higher hierarchy connectivity network and take the work of long-distance routing forwarding.

- (2) Data collected from members will be forwarded after it is fused. This mechanism reduces communication energy cost because the data traffic is reduced.
- (3) Without the need of maintaining complex routing information, member nodes have simpler function. This characteristic will further reduce the energy consumption of communication.
- (4) Clustering routing has good scalability, and it is suitable for large scale networks and distributed algorithm [13].
- (5) It is easy to overcome problems caused by mobile sensor nodes [14].

Because clustering strategy has the above benefits, it attracts extensive attention from researchers and a lot of clustering algorithms are proposed in the previous literatures. The most widely used clustering methods are LEACH [15] and its variants.

**2.2. LEACH and Its Variants.** Low Energy Adaptive Clustering Hierarchy (LEACH) is probably the most famous clustering protocol for WSN. The basic idea of LEACH is to randomly select the cluster head nodes in a circulating way and make the energy load of the entire network evenly distributed to each sensor node to achieve the purpose of reducing network energy consumption and enhancing the overall survival time of the network. Simulation result reveals LEACH can lead an additional 15% lifetime [16].

There are two procedures in LEACH clustering. In the first procedure, each node generates a random number  $\sigma \in (0, 1)$ , and if  $\sigma$  is smaller than the threshold  $T(n)$ , the node

will be selected as the cluster head. The threshold can be calculated as follows:

$$T(n) = \begin{cases} \frac{P}{1 - p[r \bmod (1/p)]}, & n \in G \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where  $p$  is the expected proportion of cluster heads in all the nodes,  $r$  is the number of current round,  $G$  is the set of nonhead nodes in the former  $1/p$  round, and  $\bmod$  is the Modulo operator. The selected head broadcasts an Advertisement Message (ADV) using Carrier Sense Multiple Access (CSMA) MAC protocol to inform that it is a cluster head. Cluster members will decide which cluster it should belong to according to its received signal strength (RSS). Then the member nodes will send Joint-REQ messages to the cluster head. Members will connect with their head using Time Division Multiple Access (TDMA) technique. The final step of the first procedure is to send a timing schedule from heads to their slave nodes.

The second procedure of LEACH is called stable procedure. In this stage, cluster members will send their perceived data to the head node within their shared time slot. In other time, member nodes will switch to sleep state for energy saving. Once the head received all the data it will make data fusion immediately to strengthen the public signal and weaken the noise. Then the head will send fused data to sink node within single hop. After the stable stage continues a period of time, LEACH will repeat the clustering stage.

Although LEACH has higher energy efficiency than many other protocols, it still has shortcomings which limits its performance. First of all, because it assumes that each node can connect with sink node directly, LEACH is not perfectly suitable for large scale networks as other clustering algorithms. Secondly, LEACH is not suitable for sensor networks with unevenly distributed energy. Thirdly, it does not define the distribution of clusters heads in the whole network; therefore, some members may not be able to find any head in their neighborhood [17]. Moreover, the single hop communication between heads and sink node may lead to a large amount of energy consumption [18].

To overcome the drawbacks above, researchers proposed many variants of LEACH, and we call them LEACH protocol family. The most important members of LEACH family include LEACH-C [19], TEEN [20], SEP [21], mobile LEACH [22], and LEACH-EEC [23]. The performance of these improved versions of original LEACH is shown in Table 1.

As shown in Table 1, no variant of LEACH can completely solve all the problems above. However, these problems are very important for WSN routing. Therefore, routing strategy of wireless sensor networks is still an open issue especially on IPv6-based routing in WSN.

Actually, LEACH protocol is quite an effective ideology in initialing clustering. Its main problem is in the later iterative clustering loops. Besides, it is not able to take full advantage of the previous clustering information. In other words, the stochastic strategy makes it difficult to make a comprehensive decision in a cluster head selection. Moreover, selecting heads before clustering may lead to an unreasonable distribution

TABLE 1: Performance of LEACH protocol family.

Protocols	Features			
	Large scale	Energy uneven	Low cost	Robustness
LEACH-C	Yes	No	Yes	No
TEEN	No	Yes	Yes	No
SEP	Yes	Yes	No	Yes
Mobile LEACH	No	Yes	No	Yes
LEACH-EEC	Yes	No	Yes	No

of cluster heads, thus increases the energy consumption. Therefore, a novel WSN routing method should be proposed, which firstly makes clustering and then selects heads in the clusters. In addition, the clustering and head selection should take many indicators into account such as node power, number of hops, RSS, head selection history, and sum of nodes.

### 3. Adaptive Clustering

**3.1. Clustering Initialization.** As the analysis in the former section, LEACH is an excellent clustering initialization method. Therefore, we use LEACH in the initial round of clustering. Thence different from later rounds of clustering, in the first round, we make head selection before clustering.

The cluster head is selected after the indicator  $\sigma$  is generated randomly. In LEACH, each slave node connects with its head in a single hop so it is enough to consider the received signal strength. Obviously, not in all cases communications in inner clusters are single hop connections. According to the research in [24], when the distance between members and their head is no longer than two hops, the cluster has the minimal energy consumption. Therefore, we modified the clustering mechanism in LEACH to constitute clusters. The number of hops is taken into account when a slave node makes decision about which cluster head it should follow. When the broadcasted ADV messages by heads are received, members will check the RSS and the number of hops.

There are three possibilities for ADV messages: (1) all single hop packets, (2) all multihops packets, and (3) mixed packets.

In case (1), member nodes choose their head according to the RSS, which is similar to LEACH.

In case (2), member nodes select their heads which could minimize the overall number of hops of the network as below:

$$\frac{\sum_{i=1}^N \{\min[h(i, 1), h(i, 2), \dots, h(i, M)]\}}{N + M}, \quad (2)$$

where  $N$  is the number of nonhead nodes in the whole sensor network,  $M$  is the number of cluster heads, and  $h(i, j)$  is the number of hops from slave node  $i$  to cluster head  $j$ .

In case (3), we introduce a lower bound of RSS to determine whether to use a single hop or multihops in communications. In most cases, the energy consumption is lower when the number of hops is less. However, when

the received signal is too weak to be processed, it may lead nodes re-send the message. Moreover, long-distance communication itself is a kind of high energy cost connection. When the RSS of a member node from a head is lower than the empirical threshold, it will be ignored. After the RSS is filtered, if only multihops ADV remains, the problem will be transformed into case (2) and the slave nodes will make their decision according to function (2). When there are still both single hop ADV and multihops ADV, member nodes will choose single hop to connect to the head as in the LEACH protocol. The schematic of clustering initialization is shown in Figure 3.

**3.2. Adaptive Iterative Clustering.** In the later rounds, different clustering strategy should be used; otherwise it is still a variant of LEACH. However, clustering information of the initial round should be used for the second round clustering. The number of clusters is a crucial factor which largely affects the energy efficiency of WSN. Unfortunately, there is no universal ideal proportion of heads in WSN clustering. In different situations, the same heads percentage may have a different performance, which can be influenced by application, data type, terrain, physical layer protocols, and so forth. Therefore, empirical percentage of cluster heads is not feasible, and to adjust the number of clusters adaptively according to the last round of clustering seems a reasonable idea.

First of all, when there are too many multihop connections and the RSS of most single hop connections is low in some clusters, it means the distance between members and the head is too long. In other words, the number of clusters is lower than that it should be in the normal state. In this case, the overlarge clusters should be split into two or more subclusters.

To evaluate whether a cluster is overlarge, the overall hops of the clusters, the average RSS, the energy cost in one round, and the amount of data transmission should be taken into account. When we assume the criteria is  $F$ , it can be calculated as follow:

$$F = \sum_{i=1}^{n+m} t(k) \frac{\sum_{i=1}^n \text{RSS}(i)}{\sum_{j=1}^m h(j) + \sum_{j=1}^{n+m} \Delta E(j)}, \quad (3)$$

where  $n$  is the number of single hop slave nodes,  $m$  is the number of multihop slave nodes,  $\text{RSS}(i)$  is the signal strength received by head from  $i$ th single hop node,  $h(j)$  is the number of hops between head and  $j$ th multihops node,  $t(k)$  is the times of sending data of  $k$ th node, and  $\Delta E(j)$  is the difference of the remaining capacity of  $j$ th node between the beginning and ending of the round.

Obviously, smaller value of  $F$  reveals a larger scale of the clusters. A threshold  $\theta$  could be used to evaluate whether the scale of cluster is reasonable or not. When  $F$  is lower than  $\theta$ , the cluster should be split.  $\theta$  is an empirical parameter which is mainly depended on the energy consumption of inner-cluster communication. It is usually between  $1.5 \sum \text{RSS}(i)$

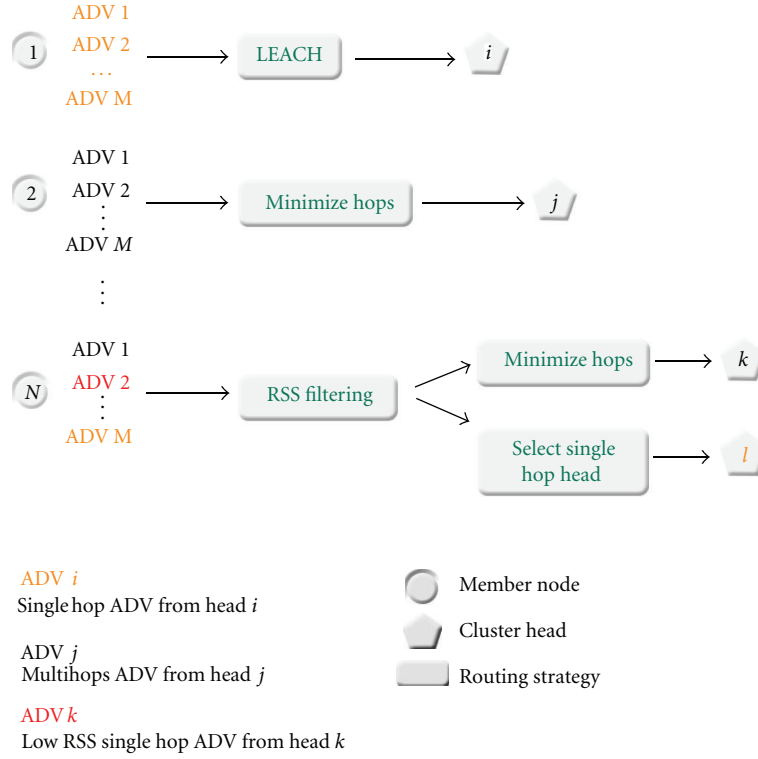


FIGURE 3: Schematic of clustering initialization.

and  $3.5 \sum \text{RSS}(i)$ . The number of split clusters could be determined by the following formula:

$$C_n = \left\lfloor \frac{\theta}{F} \right\rfloor, \quad (4)$$

where  $C_n$  is the number of clusters after the split. The cluster head selection in the split clusters is similar to that in other clusters which will be presented in the next section.

Secondly, too many clusters in the WSN will also reduce the energy efficiency of the nodes, because the nodes will be selected as clusters more frequently, when the number of clusters is larger. Thence the small clusters should be merged when the proportion of heads is too high.

Different from the *split* situation, the average number of nodes in the clusters of the entire network and the frequency of the nodes selected as heads in a certain cluster should be taken into consideration. When we assume the merge criterion of the  $i$ th cluster is  $F'(i)$ , it can be computed as below:

$$F'(i) = \frac{N \sum_{k=1}^n \tau(k)}{(N + M) \sum_{k=1}^n t(k)}, \quad (5)$$

where  $n$  is the number of nodes in cluster  $i$  and  $\tau(k)$  is the times node  $k$  has been selected as cluster head.  $N$ ,  $M$ , and  $t(k)$  have the same definition with the former functions. Similar to the split situation, a lower bound  $\theta'$  could be used to determine whether to merge the cluster with its neighbor cluster or not. When  $F' < \theta'$ , the cluster will be considered to be too small.

When a cluster head finds that the  $F'$  is lower than  $\theta'$  it will ask the adjacent heads to check whether another too-small cluster exists in its neighborhood. If there is another too-small cluster in the bordering area, the two clusters will merge and construct a new cluster. Otherwise, the too-small cluster will merge with a normal one and then be split into two clusters. The procedure is shown in Figure 4.

There is a time when the size of each cluster is appreciated but some nodes are in the wrong clusters. In this case, we need to adjust the affiliation of some nodes rather than the number of clusters. To evaluate the unsuited affiliation, the number of hops and RSS are helpful. In a multihop cluster, if the number of hops between a member and the head is significantly larger than the other slave nodes, it is better if the slave node belongs to another cluster. The significant low RSS in single hop clusters has the similar meanings. The following inequalities can be used as the criteria:

$$h(i) > \frac{n \cdot \max_{1 \leq j \leq n} [h(j)] + \sum_{j=1}^n h(j)}{2n}, \quad (6)$$

$$\text{RSS}(i) < \frac{n \cdot \min_{1 \leq j \leq n} [\text{RSS}(j)] + \sum_{j=1}^n \text{RSS}(j)}{2n},$$

where  $h(j)$  is the number of hops between the head and member  $j$  in the cluster. Other symbols have the same definitions with the former formulas. With the above inequalities, the nodes can make judgment whether its cluster affiliation is suitable or not. If the above inequalities are met, the nodes should switch to another cluster. These nodes will send a multicast request to the heads of the bordering clusters.



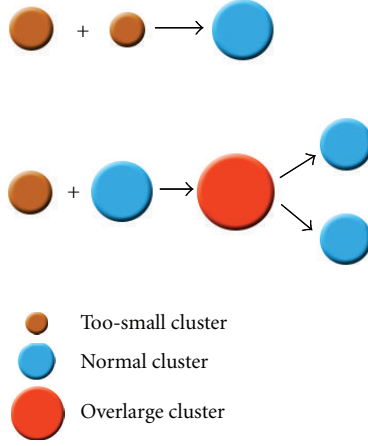


FIGURE 4: Merging the too-small clusters.

Then they will make a decision following the above principle according to the ACK of the adjacent heads:

$$C_{NEW} = \min_{1 \leq i \leq n} [RSS(i) \cdot h(i) \cdot n]. \quad (7)$$

Function (7) reveals that the nodes select the nearer and smaller cluster as its new work group in the neighborhood.

**3.3. Enable Scalability.** Scalability is a key aspect of WSN applications. In a WSN, when the proportion of the power-off nodes is higher than an upper bound, we need to add new nodes to the network. Sometimes we want to deploy more nodes to increase the coverage of the sensor network or to bring new functions. And sometimes we need to fuse two or multiple sensor networks into one network. All of the above cases need a scalable clustering algorithm.

To enable the WSN to be scalable, we need to provide the initial clustering strategies in the adaptive clustering algorithm, because the new added nodes and the old nodes are in different rounds.

Usually, there are two kinds of new added sensor nodes. Because the issues which should be addressed are different, new sensors scattered among the old nodes should be treated differently with the added sensors which are deployed spate with old nodes in a new area.

The two kinds of new added sensor nodes are as shown in Figure 5.

An important issue is how to detect the distribution of the new nodes. It is clear that when the new nodes are deployed manually, their distribution is known and could be used directly. However, in many situations, sensor nodes are sowed by airplanes or robots. In these cases, some strategies should be used to make the system detect the distribution of the new nodes automatically.

On this problem, head history is helpful because it not only reflects the history of how the nodes are selected as cluster heads but also reflects whether it is a new joined member or not. When a new joined node knows its neighbors' situation, it can detect where it is deployed.

According to the received head history, a new node may be classified into one of the following three cases.

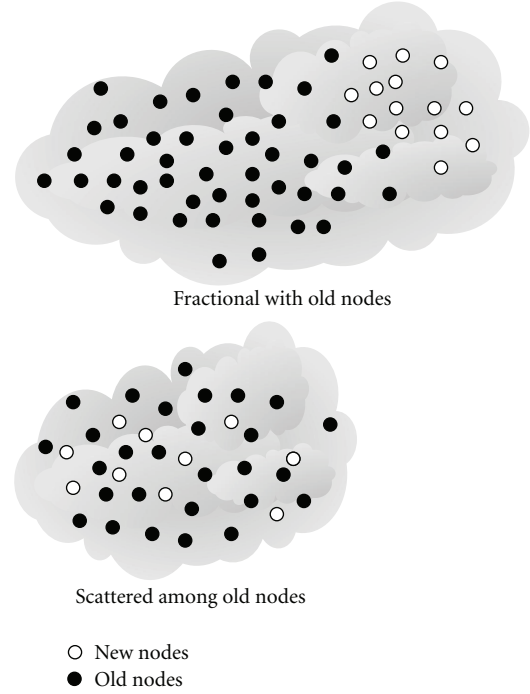


FIGURE 5: Two kinds of new added nodes distribution.

*Case 1.* All or most neighboring nodes are new joined members.

*Case 2.* All or most neighboring nodes are old nodes.

*Case 3.* In its neighboring area, new joined members and old nodes are counterparts.

The three cases above correspond to the three distribution modes of the new joined sensor nodes. Especially, Case 3 is the distribution of the boundary node in fractional distributed new nodes, and the process of distribution detection is shown in Figure 6.

The new added nodes distributed fractionally with old nodes can be considered as a new sensor network. Therefore, the strategy of its initial clustering is similar to the workflow proposed in Section 3.1. LEACH protocol is called to make the initial random head selection and sensor converge into clusters according to RSS and the number of hops.

After the clustering initialization, there is an important step named time synchronization. Time synchronization will be achieved in the second round of the new nodes by the CTSS [25] mechanism. Then the new added sensor nodes (subnetwork) can work together with the old nodes at the same rounds. During the third clustering round, the new added sensors can cluster together with the old nodes using the strategy for splitting, merging, or adjustment.

A different method is applied in the case when the new sensors are scattered among the old ones. The new nodes have to work together with the old nodes from the outset, and the initial clustering procedure of these nodes is as follows.

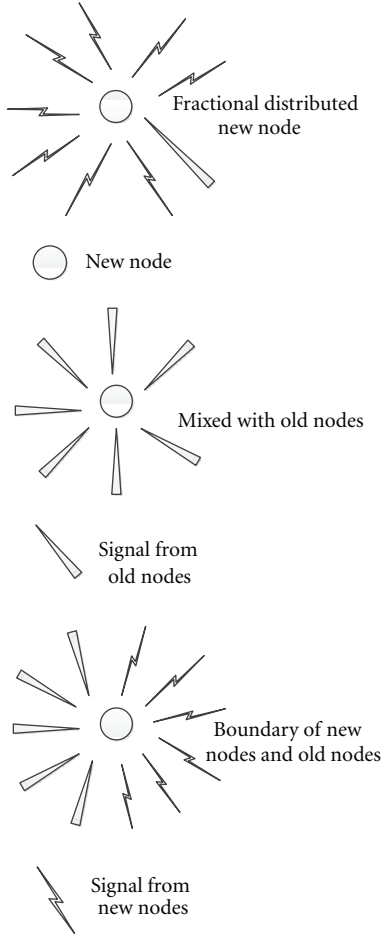


FIGURE 6: Distribution detection of the new nodes.

- (1) The new node broadcasts an ADV to all the heads of a WSN.
- (2) The ADV will be discarded when forwarded more than three hops.
- (3) Each head that received the ADV will send an ACK to the new node.
- (4) The new node will select a cluster according to the RSS or the number of hops of the ACK.

Following the steps above, the new nodes will find suitable clusters as their work groups and achieve Plug and Play. In the next round, the new nodes will be regarded as old nodes, so the clustering strategy proposed in former subsection can be deployed seamlessly. In this way, the WSN can achieve scalable clustering. No matter what the distribution of the new added nodes is, the new nodes can work and cluster together with the old sensors.

Hitherto, an adaptive clustering algorithm for WSN is proposed, which is shown in Figure 7.

This algorithm can adjust the scale of clusters and the members in the clusters adaptively according to the energy consumption in the former round. Therefore, it can achieve high energy efficiency. Because there is no missing condition in this algorithm; it is reliable and all functions are bounded.

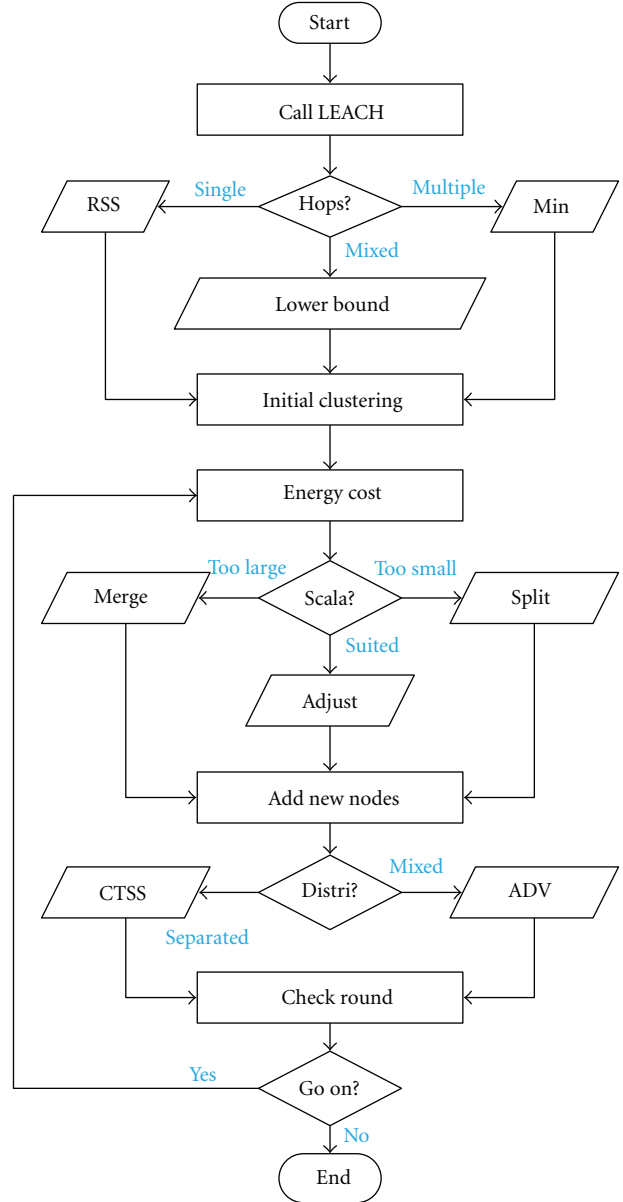


FIGURE 7: Flowchart of adaptive clustering algorithm.

Moreover, a comprehensive strategy is introduced in the algorithm to ensure its scalability.

#### 4. Adaptive Head Selection

In traditional LEACH based routing protocols, head selection is the first step of clustering. This kind of strategy is a semirandom selection method and requires that the selected heads broadcast ADV to the whole network. The semirandom mechanism makes it hard to get an optimization solution. The broadcast leads to an additional energy consumption. To solve the problems mentioned above, we design an adaptive head selection algorithm which can be used to select heads based on the principle of a minimum energy consumption after the clusters is built.

**4.1. Head Selection in Unbalanced Stage.** Unbalanced stage is the phase of the first few rounds when not every node in the cluster has been a head. At this stage, average hops and the RSS between a node and each other cluster member are unknown. So the remaining capacity of nodes and the head selection history is the most important reference for head selection.

All the nodes except the current head will be the new head candidate. They send their information to the current head. Assuming the weight of node  $i$  for head competition is  $\delta(i)$ , in the simplest case, we select the head according to the following formula:

$$\delta(i) = \frac{E_r(i)}{t_h(i) \cdot (r - r'_i)}, \quad (8)$$

where  $E_r(i)$  is the remaining capacity of node  $i$ ,  $t_h(i)$  is the times node  $i$  has been selected as a head in history,  $r$  is the current round of the cluster, and  $r'_i$  is the most recent round in which node  $i$  is a head.

The known RSS and number of hops can be regarded as the sample of overall RSS and number of hops. Image the situation that there is a lot of power in the battery of a node that has not been selected as a head for a long time, intuitively it is the ideal choice for head selection. However, this kind of node may have a long distance from the other members in the cluster. The only reason they have high remaining power is that they perceived and transported little data. Using these nodes as heads will not only enhance energy cost of the whole cluster but also make them die sooner for power exhaustion. Therefore the threshold of maximum number of hops and minimum RSS should be introduced to avoid the wrong selection of the heads. The thresholds are defined as follows:

$$\begin{aligned} \text{RSS}(t) &= \frac{\sum_{i=1}^N \text{RSS}(i)}{2n}, \\ h(t) &= \frac{2 \sum_{i=1}^{N'} h(i)}{n}, \end{aligned} \quad (9)$$

where  $N$  is the number of RSS recorded by node  $i$  and  $N'$  is the sum of the number of hops recorded by node  $i$ . When the thresholds of RSS and number of hops are taken into account, the weight  $\delta(i)$  of node  $i$  for the head selection in a cluster can be defined as

$$\begin{aligned} 0, \quad & \text{RSS}(i) < \text{RSS}(t) \cup h(i) > h(t), \\ \delta(i) &= \frac{E_r(i)}{t_h(i) \cdot (r - r'_i)}, \quad \text{otherwise.} \end{aligned} \quad (10)$$

In this way, the current head could select its successor through their weights. According to Bayesian Theory, the average duration of an unbalanced stage  $T_U$  is

$$T_U = \frac{m!}{m^k}. \quad (11)$$

The performance of impaction in unbalanced stage is analyzed latter.

**4.2. Head Selection in Balanced Stage.** When the clustering iteration runs many rounds, every node will have the experience of being a cluster head. When the head selection executes to this phase, each node in a cluster has the information of RSS and the number of hops of all the other cluster members. As we know, RSS and the number of hops have a major impact on energy consumption, so they must be considered when we make the head selection.

The average received signal strength of node  $j$  from other cluster members  $R_a(j)$  can be calculated as

$$R_a(j) = \frac{\sum_{i=1}^n \sum_{k=1}^{T_i} \text{RSS}(i, k)}{\prod_{i=1}^n T_i}, \quad (12)$$

where  $T_i$  is the times of single hop transmissions between node  $j$  and  $i$ ,  $\text{RSS}(i, k)$  is the received signal strength of node  $j$  from node  $i$ 's  $k$ th transmission, and  $n$  is the number of members in the cluster.

Similar to function (12), the average number of hops between node  $j$  and all the other nodes  $h_a(j)$  can be calculated as

$$h_a(j) = \frac{\sum_{i=1}^n \sum_{k=1}^{T'_i} h(i, k)}{\prod_{i=1}^n T'_i}. \quad (13)$$

In (13),  $T'_i$  is the number of multihop transmissions between node  $j$  and  $i$ . The definitions of other symbols are the same as (12).

In addition, the proportion  $P_M(j)$  of single hop transmissions of node  $j$  can be computed as

$$P_M(j) = \frac{\sum_{i=1}^n T_i}{\sum_{i=1}^n T_i + \sum_{i=1}^n T'_i}. \quad (14)$$

Greater number of hops means a higher energy consumption, and a higher RSS means a higher energy efficiency [26]. Therefore, formula (8) can be modified when  $R_a$ ,  $h_a$ , and  $P_M$  are taken into account. The new head selection weight of node  $j$  is

$$\begin{aligned} 0, \quad & \text{RSS}(j) < \text{RSS}(t) \cup h(j) > h(t), \\ \delta(j) &= \frac{P_M(j) \cdot R_a(j) \cdot E_r(j)}{h_a(j) \cdot t_h(j) \cdot (r - r'_j)}, \quad \text{otherwise.} \end{aligned} \quad (15)$$

Before the head selection weight calculation, nodes with the maximal  $h_a$  and minimal  $R_a$  will be excluded from the candidate list. Then the current head will select heads according to the weights of the members. Node with the highest weight will be selected as the head of the cluster in the next round.

**4.3. Deal with New Members.** The former discussion solved the problems of head selection for the old nodes in a cluster. Clusters often have new members because of the scalability of the system and the clustering strategies for *merging* and *adjustment*. There is no information available, such as average RSS, average number of hops, and percentage of single hop transmissions, for head competition. However,



sometimes new members are ideal candidates for the cluster heads. Therefore, methods should be proposed to deal with the new members of clusters.

When new members are the minority of a cluster, it is easy to be treated. Because only a few new nodes exist in the cluster, we keep them out of the head competition until next round unless they satisfy a strong constraint as below:

$$C_{\text{NEW}} = \max_{1 \leq i \leq n} [C_i], \quad (16)$$

where  $C_{\text{NEW}}$  is the remaining capacity of the new joined node and  $C_i$  is the remaining capacity of node  $i$  in the cluster. If the power of a new sensor is not the highest of the cluster, it will be excluded from the candidate list in the first round. And the unbalanced strategy will be used for head selection from the second round until each node has the experience of working as a head.

When two or more clusters merge into one cluster, the selection history of each cluster is useful for the head competition of the new merged cluster. And a two-stage head selection mechanism is used for this case. The heads in the last round before merging will select their successors, respectively, and make data exchange to compare the remaining capacity of the candidate successors. Finally the node which has the most power will be selected as the head of the new cluster.

The simplest situation is when the new nodes are deployed fractionally with the old nodes. In this case, head selection is similar to other part of the network using unbalanced strategy. After the fusion of the new jointed part, the problem is converted to the same problem described in the former paragraph.

Hitherto, the adaptive head selection algorithm is presented completely with no missing condition. It will run together with the adaptive clustering algorithm in a cross-iterative way which is called double adaptive clustering hierarchy (DACH) in this paper. The initial round of head selection is based on adaptive clustering. When the entire network is divided into clusters, the head selection algorithm will operate in each cluster. After the data transmission of the first round, the second round clustering algorithm will function based on the head selection of the first round. The iteration will be executed. The detailed workflow of the adaptive head selection is shown in Pseudocode 1.

The inner-cluster communication is achieved by DACH algorithm based on IEEE 802.15.4. Clusters will play a key role in intercluster communication based on IPv6 to make the WSN access to the internet easily and keep energy consumption in an acceptable scale at the same time.

## 5. Intercluster Communication Based on IPv6

Traditionally, WSN accesses to the internet through gateway or directly access to the internet with full IP. Access to the internet through gateway is a low energy and low computational cost solution. However, with this method, only one access point exists in the sensor network. Once the gateway fault happens, the connection between the internet and WSN will be cut off and the WSN becomes blind.

**Input:** cluster  $c = \{n_1, n_2, \dots, n_n\}$  of a sensor network

**Output:** cluster head  $h$  of  $c$

```

1 begin
2   initial head selection using LEACH
3   clustering initialization
4   for ( $i = 0, i \leq n - 1, i++$ )
5     send clustering information to current cluster head.
6     if  $n_i$  is a new cluster member of minority situation
7       make head selection according to formula (16)
8     else if  $n_i$  is new cluster member of merge case
9       goto step 2
10    else
11      unbalance? selection by formula (10): continue
12      send the full information to current head
13      head selection according to received information
14      check the timestamp
15      if ( $\text{timestamp} \leq \Delta t$ )
16        data transmission
17      else
18        clustering using adaptive clustering algorithm
19      check whether the ending of the task
20      if (not)
21        goto step 2
22      else
23        break the algorithm
24 end

```

PSEUDOCODE 1: Adaptive head section algorithm for WSN.

Moreover, single access point method tends to cause data congestion. On the other hand, full IP sensor networks will lead to huge energy consumption and greatly shorten the lifetime of WSN. To solve the problems above, we make the cluster heads communicate with the internet based on IPv6 and use IEEE 802.15.4-based inner-cluster connection.

Because the protocol header of IPv6 is overlong for the MAC protocol of WSN, the protocol stake optimization is needed for implementing IPv6 in sensor nodes.

The head structure of IPv6 is shown in Figure 1. Therefore, the compression is focused on address. IPv6 address is composed of address prefix and subnet address. The address prefix is the same when get IPv6 address using the stateless automatic address configuration. Moreover, stateless automatic address configuration achieved according to the data link address of nodes. Therefore, the header compression algorithm proposed by Junwei et al. [27] is a wonderful method. The procedures to implement IPv6 in cluster heads are as follows:

- (1) header compression using Xu's algorithm;
- (2) IPv6 package fragmentation;
- (3) communication with sink nodes, gateways, or other access points through IPv6;
- (4) IPv6 package fragments reorganization;
- (5) decompression of the received data.

Following the above steps, the cluster heads can achieve IPv6-based communication with the internet through Xu's

Achieve internet access	
IPv6 package transmission	
IPv6 package fragmentation	
Compression	Xu's algorithm
Iterative head selection	Adaptive selection
Iterative clustering	Adaptive clustering
Initial head selection	Remaining capacity
Initial clustering	LEACH
Steps	Tools

FIGURE 8: Framework of the whole system.

fragmentation algorithm. The complete framework of communication between WSN and IPV6 presented in this paper is shown in Figure 8.

As is shown in Figure 8, a novel WSN routing algorithm is proposed in this paper. The structure of sensor networks is cluster based. IEEE 802.15.4 protocol is used in inner-cluster communication to save energy and IPV6 protocol is used in intercluster communication to make the sensor network access to the internet. Comprehensive factors, such as remaining capacity, RSS, number of hops, and head history, are considered when making the clustering decision and head selection to minimize energy consumption and prolong the network's lifetime.

## 6. Simulation, Experiment, and Analysis

We make simulations and experiments to evaluate the performance of the novel algorithm. The simulation and experimental results were compared with other algorithms to get an objective and comprehensive conclusion.

**6.1. Simulations for Energy Consumption.** Lifetime of nodes is an important indicator of energy consumption. We make simulation using Matlab and compare the results of 1500 rounds with several classic WSN clustering algorithms such as original LEACH, LEACH-C, SEP, and EEF [28]. The simulation results are shown in Figure 9.

As is shown in Figure 9, DACH has the longest lifetime in the comparison group. Its energy efficiency is significantly higher than the original LEACH protocol. Moreover, after 1500 rounds of clustering, the number of survived nodes using DACH is 25 percent more than the second best algorithm.

Some WSN routing algorithms are suffering a large scale sensitive problem, which means the energy consumption of a round will increase exponentially with the increase of node number. This problem will seriously affect the performance

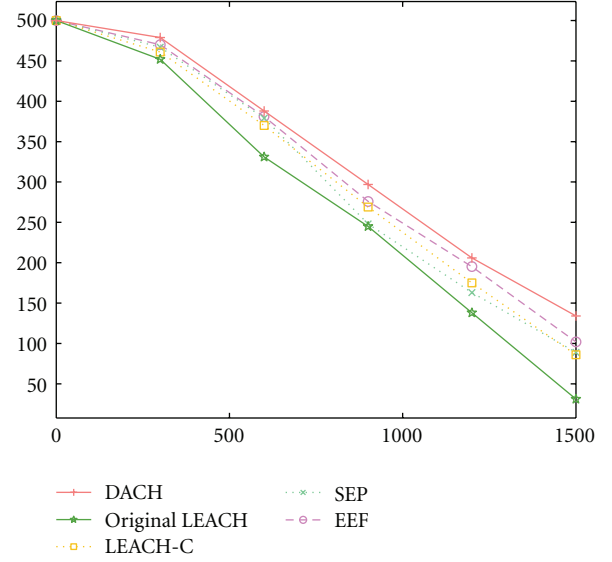


FIGURE 9: Energy efficiency of different algorithm.

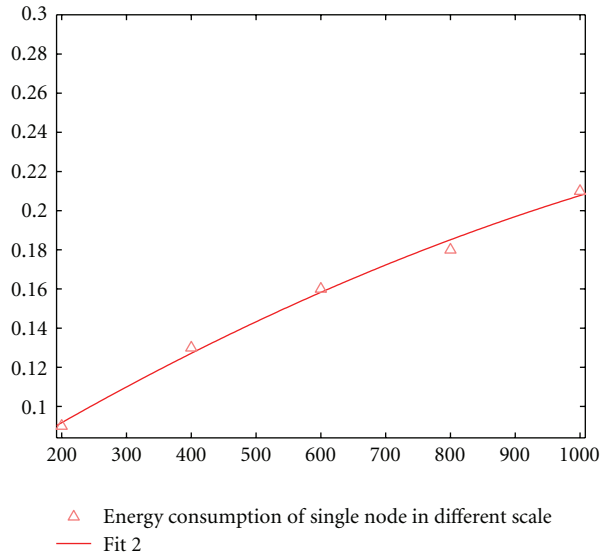


FIGURE 10: Energy consumption of single node using DACH in different sizes of WSN.

of clustering algorithm and significantly shorten the lifetime of sensor networks.

We simulate the case of energy consumption of a node sending 10 packages when the scale of the whole network is 200 nodes, 400 nodes, 600 nodes, 800 nodes, and 1000 nodes, respectively. The average energy consumption of a single node in WSN with a different scale is shown in Figure 10.

As is shown in Figure 10, the average node energy consumption grows slowly with the increase of the network scale. It means nodes using DACH algorithm do not face a large scale sensitive problem. The algorithm is reliable with different size.

Another important issue that should be discussed is the performance impaction of an unbalanced stage which

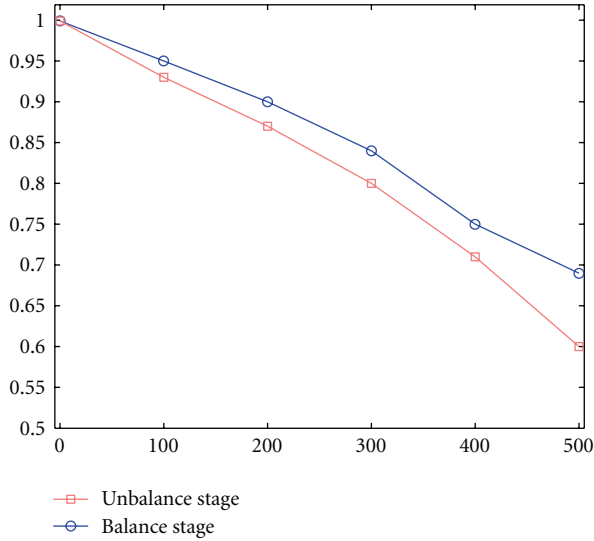


FIGURE 11: Impact of unbalanced and balanced stages.



FIGURE 12: Sensor node used in the experiment.

is analyzed in Section 3.3. To evaluate its impact, we simulated the remaining battery capacity in balanced and unbalanced stages after 100 rounds, 200 rounds, 300 rounds, 400 rounds, and 500 rounds, respectively. The simulation results are shown in Figure 11.

Figure 11 reveals that the energy consumption of the unbalanced stage is higher than that of the balanced stage. After 500 rounds, the nodes in the unbalanced stage have a capacity nearly 10% less than nodes in the balanced stage. An alternative strategy is making the  $n$  nodes of the cluster be heads ordinarily in the first  $n$  round.

**6.2. Experimental Results and Analysis.** We use 50 sensor nodes to form a real-world system using IRIS. Each of these node uses two 1.5 V, 1500 mA AA batteries. The picture of sensor nodes is shown in Figure 12.

We make the sensor network connect with the internet in different ways including full IP connection, static heads connection, LEACH-based connection, and DACH-based

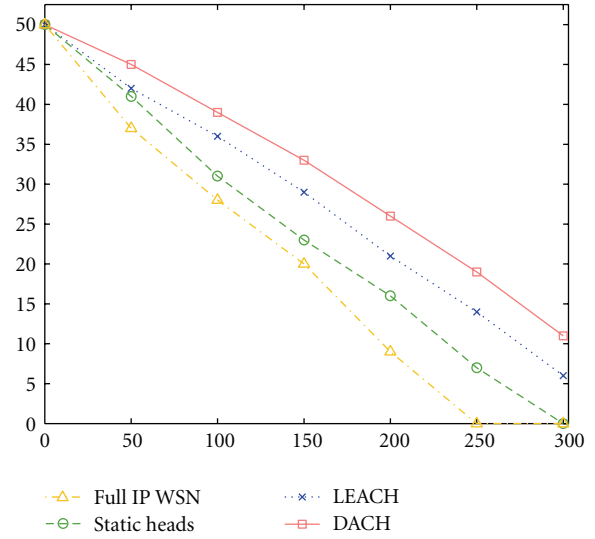


FIGURE 13: Lifetime of different algorithms.

connection. Each algorithm was tested by 300 rounds. Each round contains 10 measurements. Therefore, each node sends 3000 packages. The experimental results are shown in Figure 13.

Figure 13 reveals that DACH has the lowest energy consumption. WSN based on DACH which contains 50 sensor nodes has more than 100 rounds longer lifetime than the full IP-based routing.

## 7. Conclusion

IPv6 may bring a lot of benefits to sensor networks because of its high security, high robustness, mass amount of address, and so forth. However, the high energy cost of IPv6-based communication makes it difficult to be used in WSN. This paper proposed a double adaptive clustering hierarchy algorithm. After LEACH is called for clustering initialization, DACH will make clustering and head selection adaptively according to RSS, number of hops, remaining capacity, and head selection history. The comprehensive consideration which takes all energy cost factors into account leads to high energy efficiency and makes the energy consumption distribution even more. Therefore, the nodes and the whole sensor networks based on DACH can achieve access to the internet through IPv6 with a significantly longer lifetime.

The mobility of this algorithm is not verified. In addition, package loss rate should be further reduced. This will be undertaken as a future work on this topic.

## Acknowledgment

This research was supported by CNGI.

## References

- [1] M. B. Krishna and M. N. Doha, "Self-organized energy conscious clustering protocol for wireless sensor networks," in

- Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT '12)*, pp. 521–526, March 2012.
- [2] L. D. Paulson, "Will wireless be IPv6's killer app?" *Computer*, vol. 34, no. 1, pp. 28–29, 2001.
  - [3] E. Lee, S. Park, F. Yu, and S. H. Kim, "Data gathering mechanism with local sink in geographic routing for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1433–1441, 2010.
  - [4] W. Xiaonan and G. Demin, "An IPv6 address configuration scheme for All-IP wireless sensor networks," *Ad-Hoc and Sensor Wireless Networks*, vol. 12, no. 3–4, pp. 209–227, 2011.
  - [5] A. Dunkels, J. Alonso, and T. Voigt, "Making TCP/IP viable for wireless sensor networks," in *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, pp. 1–9, November 2003.
  - [6] P. A. C. S. Neves, A. Esteves, R. Cunha, and J. J. P. C. Rodrigues, "User-centric data gathering multi-channel system for IPv6-enabled wireless sensor networks," *International Journal of Sensor Networks*, vol. 9, no. 1, pp. 13–23, 2011.
  - [7] W. Xiaonan and Q. Huayan, "An IPv6 address configuration scheme for wireless sensor networks," *Computer Standards and Interfaces*, vol. 34, no. 3, pp. 334–341, 2012.
  - [8] Y. A. Liu, L. Ye, Q. M. Shao, and B. H. Tang, "Integrating wireless sensor networks with the TCP/IP networks," *Journal of Beijing University of Posts and Telecommunications*, vol. 29, no. 6, pp. 1–5, 2006.
  - [9] H. Khaleel, F. Penna, C. Pastrone, and R. Tomasi, "Frequency agile wireless sensor networks: design and implementation," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1599–1608, 2012.
  - [10] J. Aweya, M. Ouellette, D. Y. Montuno, and K. Felske, "Circuit emulation services over Ethernet—part 1: clock synchronization using timestamps," *International Journal of Network Management*, vol. 14, no. 1, pp. 29–44, 2004.
  - [11] J. Wan and Q. Liu, "Efficient data association in visual sensor networks with missing detection," *Eurasip Journal on Advances in Signal Processing*, vol. 2011, Article ID 176026, 2011.
  - [12] L. Xiaojuan, Y. Huiqun, and W. Xiang, "Energy-aware topology evolution model with link and node deletion in wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2012, Article ID 281465, pp. 1–14, 2012.
  - [13] M. Youssef, A. Youssef, and M. Younis, "Overlapping multihop clustering for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1844–1856, 2009.
  - [14] Z. Quan, A. Subramanian, and A. H. Sayed, "REACA: an efficient protocol architecture for large scale sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2924–2933, 2007.
  - [15] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS '00)*, p. 223, January 2000.
  - [16] A. Nayeibi and H. Sarbazi-Azad, "Performance modeling of the LEACH protocol for mobile wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 812–821, 2011.
  - [17] J. Jinbao and Z. Lijun, "An advanced LEACH protocol for wireless sensor networks," in *Proceedings of the International Conference on Modeling, Simulation and Optimization*, pp. 346–349, December 2009.
  - [18] X. Gao, Y. Yang, and D. Zhou, "Coverage of communication-based sensor nodes deployed location and energy efficient clustering algorithm in WSN," *Journal of Systems Engineering and Electronics*, vol. 21, no. 4, pp. 698–704, 2010.
  - [19] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S8–S13, 2005.
  - [20] A. Manjeshwar and D. Agrawal, "TEEN: a routing protocol for enhance efficiency in wireless sensor networks," in *Proceedings of 15th International Parallel and Distributed Processing*, pp. 26–35, August 2001.
  - [21] G. Ragdakis, I. Matta, and A. Bestavros, "SEP: a stable election protocol for clustered heterogeneous wireless sensor network," in *Proceedings of The 1st International Workshop on SANPA*, pp. 107–115, March 2004.
  - [22] A. Papadopoulos, A. Navarra, J. A. McCann, and C. M. Pinotti, "VIBE: an energy efficient routing protocol for dense and mobile sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1177–1190, 2012.
  - [23] Asaduzzaman and H. Y. Kong, "Energy efficient cooperative LEACH protocol for wireless sensor networks," *Journal of Communications and Networks*, vol. 12, no. 4, pp. 358–365, 2010.
  - [24] M. Horauer, K. Schossmaier, U. Schmid, and R. holler, "PSynUTC-evaluation of a high-precision time synchronization prototype system for Ethernet LANs," in *Proceedings of the 34th IEEE Precise Time and Time Interval Systems and Application Meeting*, pp. 263–277, December 2002.
  - [25] A. S. Hu and S. D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2725–2748, 2006.
  - [26] T. Guodong, Z. Kougen, and Y. Ge, "A mobile-beacon-assisted sensor network localization based on RSS and connectivity observations," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 487209, 14 pages, 2011.
  - [27] X. Junwei, Z. Baohua, and Q. Yugui, "Research on IPv6 routing algorithm in WSN," *Journal of University of Science and Technology of China*, vol. 37, no. 2, pp. 1534–1538, 2007.
  - [28] H. Jiang, S. Jin, and C. Wang, "Framework for clustering-based data collection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 1064–1071, 2011.



